UNIVERSITY OF SOUTHAMPTON

Applying Gamification to Teaching Cyber Security

By

Reece Buckle

Project Supervisor: Dr Nawfal Fadhel

Second Examiner: Dr Andrew Sogokon

A project progress report submitted for the award of BSc Computer Science

Department of Electronics and Computer Science

Novemeber 2020

Abstract

Due to the ever changing landscape of technology and cyber security, there is a requirement to promote cyber security awareness where possible. Typically this is done via government endorsed training schemes, website campaigns, fliers and posters and gamified strategies. As a strategy towards this problem, this report investigates the use of gamification mechanics in order to teach cyber security concepts in a fun and interactive way. Furthermore, despite many serious cyber security games being developed for Universities and businesses, there are a lack of cyber security games that are also relevant and available to the general public. Therefore, this project will aspire to develop a multiplayer board game in which cyber security vulnerabilities are illustrated through mini-game challenges, cyber attack & defence cards and a contextually appropriate setting. The goal of this project is to produce a game which is relevant for both businesses and the public domain - and ultimately leaves the player more conscientious in regards to how they interact with technology.

Statement of Originality

I have read and understood the ECS Academic Integrity information and the University's Academic Integrity Guidance for Students.

I am aware that failure to act in accordance with the Regulations Governing Academic Integrity may lead to the imposition of penalties which, for the most serious cases, may include termination of programme.

I consent to the University copying and distributing any or all of my work in any form and using third parties (who may be based outside the EU/EEA) to verify whether my work contains plagiarised material, and for quality assurance purposes.

I have acknowledged all sources, and identified any content taken from elsewhere.

I have not used any resources produced by anyone else.

I have used some royalty-free assets for my game (few sound affects, the tile pattern, 3D Unit Models and the UI icons. These are acknowledged during the report of mention, as well as in the acknowledgements of my game - should I make it doubly clear here as well?

I did all the work myself, or with my allocated group, and have not helped anyone else.

The material in the report is genuine, and I have included all my data/-code/designs.

I have not submitted any part of this work for another assessment.

Contents

1 Introduction

As modern-day technology is ever evolving, the number of users who interact with technology increases consequently. As a result, the risk of an individual, or business, becoming a victim to cyber-crime increases proportionately. Small and medium-sized businesses (SMBs) are the biggest sectors targeted by cyber-criminals [1], which stem from issues such as budget restraints and expressing a lack of understanding towards cyber security concepts.

In fact, because of COVID-19 changing the dynamic of industry standards this year, a statistical analysis from May 2020 (UK) showed that individuals experiencing targeted hacking increased by 77.41% - in comparison with the previous year [2]. This is most likely since employees are encouraged to work from home via their personal computers. Consequently, this has fed into a new strategy whereby cyber-criminals are moving laterally into organisational infrastructure by targeting and infecting employees at their less secure personal computers [3].

Regarding this problem, this paper will explore the effectiveness of educational games - which has been shown to have an advantage on the learning outcome in comparison with traditional training material [4].

Problem Statement

Despite the existence of many cyber security awareness programs, there is still a lack of effective, widespread cyber security training

Research Question

Does teaching cyber security through a gamified medium improve user confidence in protecting against cyber-attacks?

Hypothesis

Creating an educational cyber security game will leave users feeling more confident and aware with regards to cyber security concepts

1.1 Goals

The primary goal is to apply gamification mechanics, to a multi-player game, in order to illustrate the OWASP Top 10 security weaknesses. The expected result of this project is to create an accessible, online, multiplayer board game which could be used for both recreational and training purposes.

Application specific goals:

- The application will support 2-person multiplayer
- The application will be accessible on PC, tablets and the web
- The application will be playable cross-platform
- The application will illustrate a range of cyber security attacks and weaknesses

- The application will be extensible to support new scenarios and mechanics

1.2 Scope

As cyber-security is an incredibly broad field, only the OWASP Top 10 will be used to create an educational scenario of defending and attacking web applications. During the literature review, a range of pre-existing online game types are evaluated – with appropriate mechanics extracted and identified from Forde's research [5]. Using the findings from this evaluation, and the MDA framework [6], an appropriate serious game style is identified – with multiplayer being the core focus. Finally, the application is completed using Unity3D, Photon Unity Networking 2 and Photon Cloud.

2 Literature Review

This section aims to research the current problem with educational cyber security training methodologies to design an application which fulfils these requirements. As such, a range of pre-existing games, gamification mechanics and frameworks are evaluated.

2.1 The Problem with Cyber Security Training Programs

Regarding the delivery of training programs, providing generalised cyber security advice has little effect on changing the behaviour of employees within SMBs as they fail to relate the advice to their own workplace-environment [1, 7]. Furthermore, non scenario-specific training fails to illustrate how security risks link together in a multifaceted social engineering attack [7]. Lastly, training programs are generally undertaken in a formal setting which fosters a situation of recipients not absorbing the information effectively as they aim to finish the material as fast as possible [7].

To summarise, the 4 core problems for SMBs [1, 8], are:

1. SMBs can be heavily constrained by a limited budget
2. SMBs can be difficult to reach as they do not understand the severity of data breaches
3. SMBs are often distracted by the operational requirements for setting up and running a small business
4. SMBs struggle to identify their assets in terms of the risks associated with them

From this, it is evident that a light-weight training application that can be tailored for multiple scenarios would be an appropriate solution for training these employees. Furthermore, delivering the training material through a gamified medium should lead to an increased rate of information absorption - even in a workplace environment.

2.2 Why Use a Game-Based Learning Approach?

Many of the challenges from traditional training programs can be overcome with game-based learning strategies [4]. A summary of key strengths and motivations for these strategies [4, 9, 10, 11, 12], include:

- Rapid progress paired with instant feedback
- Strong user engagement
- Allows the user to learn from mistakes in a safe, non punishing environment which would otherwise discourage exploration (due to the fear of failure)
- Serves as a platform to encourage self-learning at the user's desired pace

- Allows the user to become deeply immersed into the 'game - world' where learning feels like a secondary objective
- Relatively cheap and low cost to produce in comparison to larger training schemes
- Requires little to no supervision
- Easy to distribute across multiple platforms and incorporate into the workplace
- Easy to integrate within events such as hackathons and other cyber security awareness gatherings
- Use of an integrated rewards system such as badges, hidden achievements and the desire to win
- Educationally appropriate by incorporating a structure/story that is contextually similar to the real world

2.3 Appropriate Gamification Mechanics

In order to evaluate pre-existing cyber security games in the next section, Forde's report [5] identifies the following gamification mechanics which increase the adoption rate of positive cyber security trends within the workplace:

- Avatar & User Profile
- Badges & Privileges
- Challenge
- Competition
- Collaboration
- Feedback & Guidance
- Goals & Objectives
- Incentives & Rewards
- Leaderboards
- Points System
- Progress & Levels
- Role Playing
- Story
- Tips & Hints System

2.4 An Analysis of Pre-Existing Cyber Security Games

Still formatting this table and need to redo the analysis after, and map them into this project

Game	Game Type	Description & Key Findings	Target Audience	Mechanics
Game of Threats	Multiplatform (Mobile, PC)	Employees are split into teams of attackers and defenders who work together to simulate scenarios of cyber attacks and appropriate responses	Business Employees	Feedback Incentives Competition
Webonauts Internet Academy	Web Application Point and Click Side Scroller	Puts the player as an astronaut in which they can rank up their status by demonstrating smart and good behaviour	Children (aged 7-12)	Avatar Feedback Tips / Hints Badges
Cybersecurity Lab	Web Application Point and Click	Allows the player to choose a business they'd like to start and require them to spend defence points in different areas of cyber defence.	Children (aged 7-12)	Avatar Achievements Progress Point System Tips / Hints Feedback
Targeted Attack	Web Application Point and Click	Targeted Attack places you as a CEO in a simulation of business growth and defence from cyber attacks	Business Employees	Feedback Story Challenge
Keep Tradition Secure	Web Application Point and Click	You are a campus student trying to take down a fictional cyber criminal by making smart cyber security decisions	University Students	Tips / Hints Feedback Rewards
Classcraft	Web Application Point and Click	Classcraft incorporates gamifiication principles through the use of management software to set goals and challenges within a classroom and encourages teamwork between students	Students (Lower Education)	Avatar Leaderboard Competition Badges Feedback Goals Incentive Point Systems
Cyber Awareness Challenge	Training Simulation	Single Player simulation of everyday life within the workplace and how to behave safely and responsibly	Business Employees	Tips / Hints Feedback Story Points System
Cyberland - Cyber Security Challenge	Web Application Point and Click	Cyber Security Challenge UK is an organisation which hosts a variety of mini games (Cyberland), competitions and networking between schools, universities, businesses and government institutes	Students (All Levels)	Competition Feedback Tips / Hints Story Goals
Hacknet	Video Game Point and Click	Hacknet is a paid game (on Steam) which is a terminal-based hacking simulator	Gamers	Story Progress Feedback Achievements

2.5 How This Project Stands Out

An analysis and specification of the solution to the problem, as well as meeting all the requirements of reviewing pre-existing games

3 Requirements

Based on the findings in the literature review, appropriate personas are constructed from the stakeholders who could benefit from the developed application. From this, requirements are identified and refined into user stories. These user stories are then prioritised according to the MoSCoW analysis and distributed into a Gantt Chart for a visual representation.

3.1 Stakeholder - Personas

From analysis of pre-existing cyber security games, a list of stakeholders in which educational cyber security games were developed and targeting includes:

- Students (Higher Education)
- Researchers and Professors
- Employees (SMBs)
- Security Consultants
- Security Advocates

Figure 1: Royalty - Free Image [13]

Persona One - Olivia - Student

Olivia is an undergraduate Physics student with a strong interest in programming and web development. Although she does not major in Computer Science, she would still like to learn more about the most common web application security risks.

Olivia would also love to educate her parents in basic cyber-security as her own father is not well acquainted with internet safety – and has succumbed to phishing emails himself.

Because her Dad suffers from colour blindness, her requirements is a colour-blind friendly game in which she could learn the rules and play with her father.

In this scenario, Olivia could benefit from the application by playing out multiple scenarios with her father – that would both inform her about secure web-development, as well as engaging her father with the world of cyber-security.

Figure 2: Royalty - Free Image [14]

Persona Two - Ben - Software Consultant

Ben is a software consultant for a medium-sized business. Although he has a degree in Computer Science, he is not an expert in cyber security.

As a consultant, Ben would like to develop software for his clients that is as secure as it is usable. Since this is done in a team, Ben would like to host an activity which brings together his co-workers (of varying experience) to participate in keeping up to date with current security trends.

His requirements are free, lightweight training material as he does not have the required funding, or time, to hire an expert security consultant.

To benefit his goal, the application will have multiplayer that will encourage his employees to challenge each-other in a stimulating way. Furthermore, the gamification aspect will not require his co-workers to work hard outside of their contracted hours.

Figure 3: Royalty - Free Image [15]

Persona Three - Samira - Security Analyst

Samira is a senior level security analyst. Samira completed her PhD on using gamification to teach security.

As an expert in her field, her passion for cyber security extends to all her colleagues, friends, and family. As such, she has a strong desire in raising awareness and eliminating the human error which lead to most security exploits.

Samira would benefit from the application by using it as a framework to map the mechanics to alternative cyber-risk scenarios, thus creating a training tool. Furthermore, she could use the application as a foundation to further her own research in gamification techniques.

3.2 Functional Requirements

The following requirements are derived from the findings in the literature review, as well as specific features that proposed application should have.

ID	Requirement	Description	MoSCoW
1	Multiplayer	Multiplayer turns between connected users	Must
2	Matchmaking	Users can host and join a game session	Must
3	Quit / Leave	Users can leave at any point	Must
4	Web Accessible	Playable on a web browser	Should
5	View Units	Users can see the status of all units	Should
6	View Description	Units and moves have an explanation	Must
7	View History	Users can see the the game sequence	Must
8	Complete	Users can play a full version of the game	Must
9	Clear Goal	Gameplay loop and objectives are clear	Must
10	Single Player	Users can play single player	Won't
11	Movement	Units move with varied distances correctly	Must
12	Unit behaviour	Units have unique stats, moves and abilities	Must
13	Tutorial	Users can learn the rules and mechanics easily	Must
14	Gamification	Gamification mechanics should encite learning	Must
15	Challenge	The game should invoke critical thinking	Must

Table 1: Functional Requirements

3.3 Non - Functional Requirements

The following requirements outline how the application should perform. Although these are not required to reach a minimum viable product, they benefit the effectivity of the application greatly.

ID	Requirement	Description	MoSCoW
16	Availability	Cross-platform playable (e.g. Tablet browser)	Could
17	Ease of use	Easy to learn, understand and use	Should
18	Accessibility	Compliant with WCAG 2.1 and colour-blind friendly	Should
19	Secure	Playable on a secure website domain	Should
20	Correct	Factually correct in any cyber security concepts explored	Must
21	Performance	Gameplay should be 30 fps with low multiplayer latency	Must
22	Scalable	Scalable to support more than 2 players at any given time	Should
23	Learning	Users can learn deeper cyber security principles	Must
24	Extensible	Flexibility to add different scenarios, units and abilities	Won't
25	Feedback	Provides custom feedback based on performance (e.g. Chess)	Could
26	Music	Good music that fits the theme and aesthetic	Should
27	Animations	Unit movement, and abilities, have satisfying animations	Should
28	Enjoyable	Depth of mechanics and dynamics that lead to meaningful play	Must

Table 2: Non-Functional Requirements

3.4 User Stories

ID	As a (role/user)	I want a/to.. (feature)	So that I can... (benefit)	MoSCoW	Ref	Difficulty
1	General Player	View a unit's stats (HP/Defence)	Know who to attack	Must	5,6	Low
2	General Player	View an ability's range, damage & cost	Know what I can do	Must	5	Low
3	General Player	View a colour-blind friendly UI	Interact with the application easily	Should	17	Low
4	General Player	View a unit's movement range	Where I can move to	Must	11	Medium
5	General Player	View a unit's ability attack range	Know who I can reach	Must	12	Medium
6	General Player	Move a unit	Attack a target	Must	8,12	High
7	General Player	Attack a target	Kill an enemy unit	Must	8,13	Medium
8	General Player	Disable the target	Prevent them from moving	Must	8,14	Medium
9	General Player	Defend an ally unit	Prevent them from dying	Must	8,15	Medium
10	General Player	Click on a move description	Learn more about the attack nature	Must	6	Low
11	General Player	Click on a unit	Learn more about the attack vector	Must	7	Low
12	General Player	Click on and view a history log	Follow the game's sequence of events	Should	7	Low
13	General Player	Click on a unit map	Know how units relate to each other	Could	26	High
14	General Player	View the units remaining	I know what units I can still move	Should	5	Low
15	General Player	Pan the camera and/or zoom in	See the map from different angles	Could	20	Medium
16	General Player	Unique animations for each ability	Differentiate abilities	Could	24,28	Very High
17	General Player	Varied unity abilities	Enjoy varied gameplay	Could	12,24	High
18	General Player	An intermediate tutorial	Quickly learn how to play	Should	13	Very High
19	General Player	A help / status bar	Understand the gameplay	Should	13,23	Medium
20	General Player	Multiplayer functionality	Host and join a session	Must	1,2,8	Very High
21	General Player	Single player functionality	Play solo	Won't	10	N/A
22	General Player	Quit / leave the session	Play with a friend	Must	3,8	Low
23	General Player	Win or lose	Finish a game session	Must	8	Low
24	General Player	Play on Desktop	Play on my PC	Must	14	Medium
25	General Player	Play on Web Browser	Play on my browser	Should	4, 14	Medium
26	General Player	Play on Tablet	Play on my tablet	Should	14	Medium
27	Security Analyst	Play cross-platform sessions	Organise fun, training sessions	Could	1,2,14	High
28	Student	Learn about the OWASP #10	Understand attacks and mitigations	Must	14,23	Low
29	General Player	Purchase & upgrade new units	Have a deeper reason to keep playing	Could	12,25	Very High
30	Security Analyst	Receive feedback after a game	Know how to play better next time	Could	26	Very High
31	General Player	Hear gameplay music	Become immersed within the game	Should	24,27	Medium

Table 3: Identifying User Stories from the Personas and Application Requirements

3.5 Constraints and Limitations

- The application has to be optimised with simple textures to run on a web browser due to rendering limitations in WebGL
- Different web browsers have different rendering abilities
- The application will have resizing and mouse interactivity limitations on different devices
- Given the development time limit, only one game "scenario" will created.
- It will not be possible to iteratively develop a game after sufficient play-testing and receiving user-feedback

- Due to Covid19, it will not be possible to create a physical paper prototype of the application, and evaluate this beforehand

4 Design

This section gives a top-level overview of the physical design, and the theory behind each implementation of the project.

4.1 Applying the MDA Framework

The MDA (Mechanics, Dynamics, Aesthetics) framework [6] is the most well-known framework to identify the requirements of building a game. It is an iterative process in which the mechanics of the game are first outlined, and evaluated, to see what game-play loops (dynamics) emerge. Finally, the dynamics are mapped to a target aesthetic– e.g. the scarcity of resources is an important dynamic for a survival title.

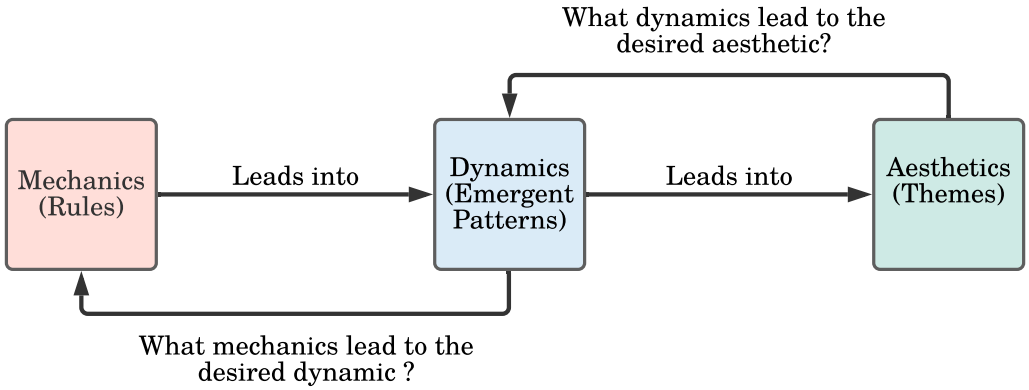


Figure 4: The Mechanics-Dynamics-Aesthetics Model

4.1.1 Mechanics

The mechanics are fundamentally the most basic rules and actions a player can do within the game world [6]. They can be identified as follows:

- Action Point (AP) requirement for certain games
- Moves limited by a distance range and can be blocked by other
- Movement opportunities is blocked by both enemy and friendly units
- Units can only attack once and move once per turn
- Units can do damage, bypass shields, restore defence and disable other units
- Disabling a database / web server unit will stop AP gain from that unit for one turn
- Disabling movable units will prevent from acting for one turn
- Units have a max defence threshold cannot be restored past the threshold

- When a unit has 0 HP, it will be removed from the game world
- Units cannot move after attacking (even if they did not move that turn)
- Each movable unit has 2 abilities, whereas the static units (database / web server) cannot directly be interacted with

4.1.2 Dynamics

The dynamics can be thought of as 'the emergent behaviour that arises from gameplay when the mechanics are implemented' [6].They can be identified as follows:

- Players may choose to target the web server/database first to reduce AP gain
- Players may use shield destroying moves on heavy units
- Players may use cheaper moves to save AP
- Players may block their database/web server to protect them
- Players may use the bypass defence move to target units with a lot of defence but little HP (e.g. web server)

4.1.3 Aesthetics

The aesthetics can be thought of as 'the emotional response a game should illicit from the player' [6]. They can be broken down into:

- Sensation (emotion-invoking)
- Fantasy (immersion)
- Narrative (story - rich)
- **Challenge (puzzles / obstacles)***
- **Fellowship (co-operative / multiplayer)***
- Discovery (open world)
- Expression (character creation)
- Submission (simulations)

* Where Fellowship and Challenge are the two key target Aesthetics for the purpose of this application.

4.2 Mapping the OWASP Top 10 to Game Mechanics

Figure 5 represents the choice of mapping security risks, exploits and mitigations to in-game mechanics. The nature of these mechanics are explored in section **??**. The OWASP Top 10 was chosen as it represents a broad consensus of the most common web application security risks [16].

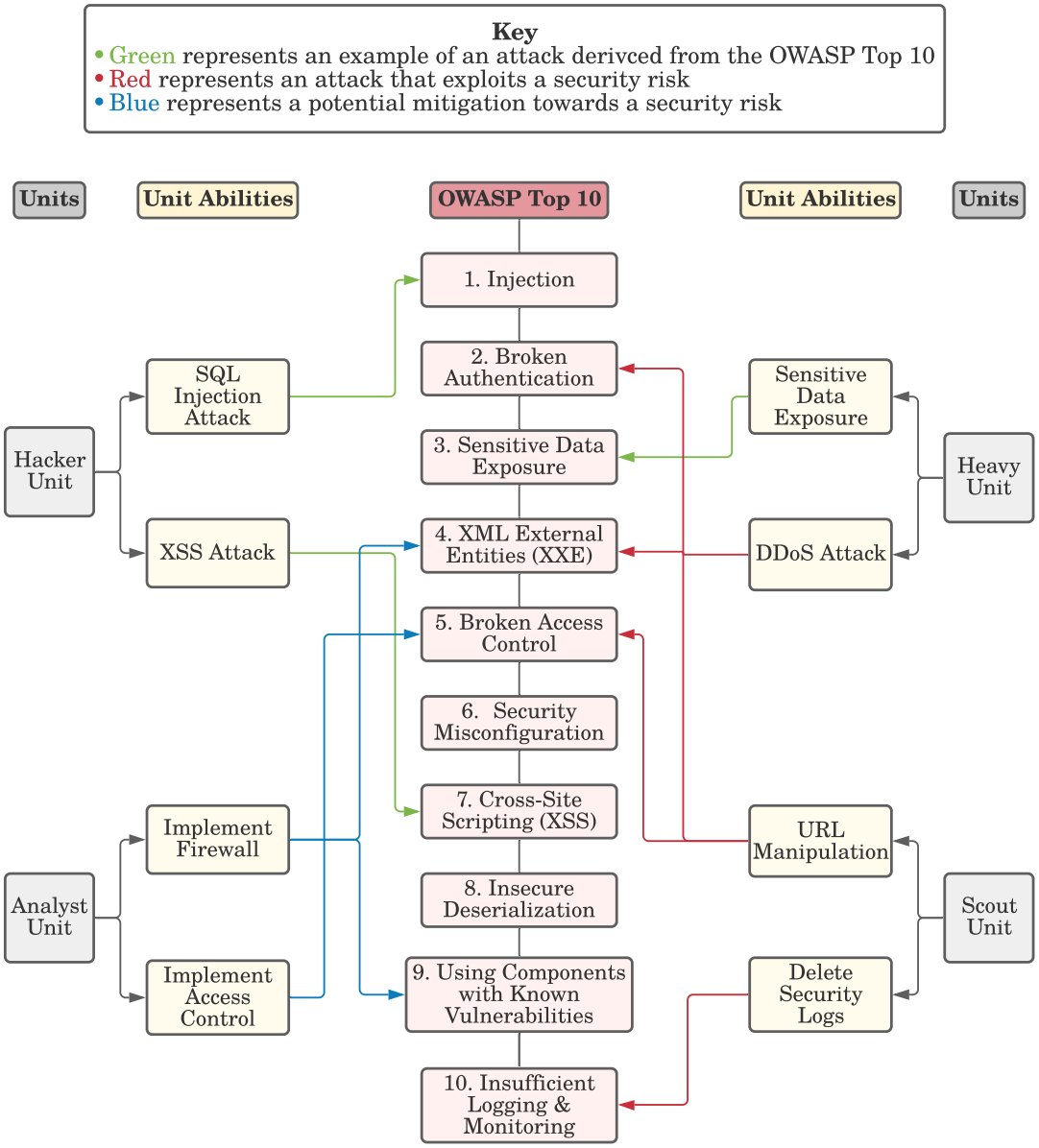


Figure 5: Mapping the OWASP Top 10 to Game - Mechanics

Need to summarise the content in this table and reference properly

Ability	Description	ID
SQL Injection	Sequel Query Language (SQL) injection exploits the interactivity between the web application and commands that query a backend database. Using SQL injection, it is possible to bypass authorisation, extract sensitive information from databases and even manipulate database entries with your own custom information.	1
XSS Attack	Cross-Site Scripting (XSS) attacks can occur when a web application fails to validate user input by escaping special characters. As a result, an attacker can execute scripts in a victim's browser, hijack their session data and even redirect them to other phishing sites.	7
Firewall	Firewalls block and filter unwanted traffic to keep the web application isolated within a safe environment. Furthermore, firewalls can be used in conjunction with an Intrusion Detection/Prevention system to detect malicious activity and send alerts upon detection. A firewall can also be used to mitigate XML External Entities attacks (XXE)	4,9
DDoS	A DDoS attack involves using a network of devices to flood a web application, network or server with overwhelming internet traffic - and disable it from being able to handle requests. Although not directly listed in the OWASP Top Ten, Denial of Service attacks can be accomplished from exploiting XML External Entities (XXE)	4
URL Manipulation	URL Manipulation changes the URL parameters to bypass authorisation in websites that suffer from broken access control. The goal is to access admin locations, root files and sensitive data (such as software versions) which could be used to identify vulnerabilities within the system.	2,5
Sensitive Data Exposure	Examples of sensitive data include personal, account and financial information. Improper handling of sensitive data in transit or storage lead to data breaches which can cripple a company's reputation. In more severe outcomes, such information could be used to commit fraud and identity theft.	3
Delete Security Logs	Insufficient data logging and monitoring allows an attacker to persist in a compromised system - undetected for a longer time without being detected. Logging pertains to recording anomalies, failed login attempts, user activity and so on, whilst effective monitoring is to know how to interpret the logged data - and how to react appropriately in the case of such detection.	10
Access Control Systems	Broken access control is present when employees/users have access to higher levels of clearance, sensitive information and possibly even "admin" operations than they require.A good access control system should be implemented once, and periodically reviewed. Control policies based on user roles should enforce the least amount of privileges each user requires to complete their interaction within the web application.	5

Table 4: A Description of Ability Names

4.3 UML Modelling - A Class Diagram Representation

Figure 6 represents the core basis of functional code within the application but omits all testing, utility and helper scripts.

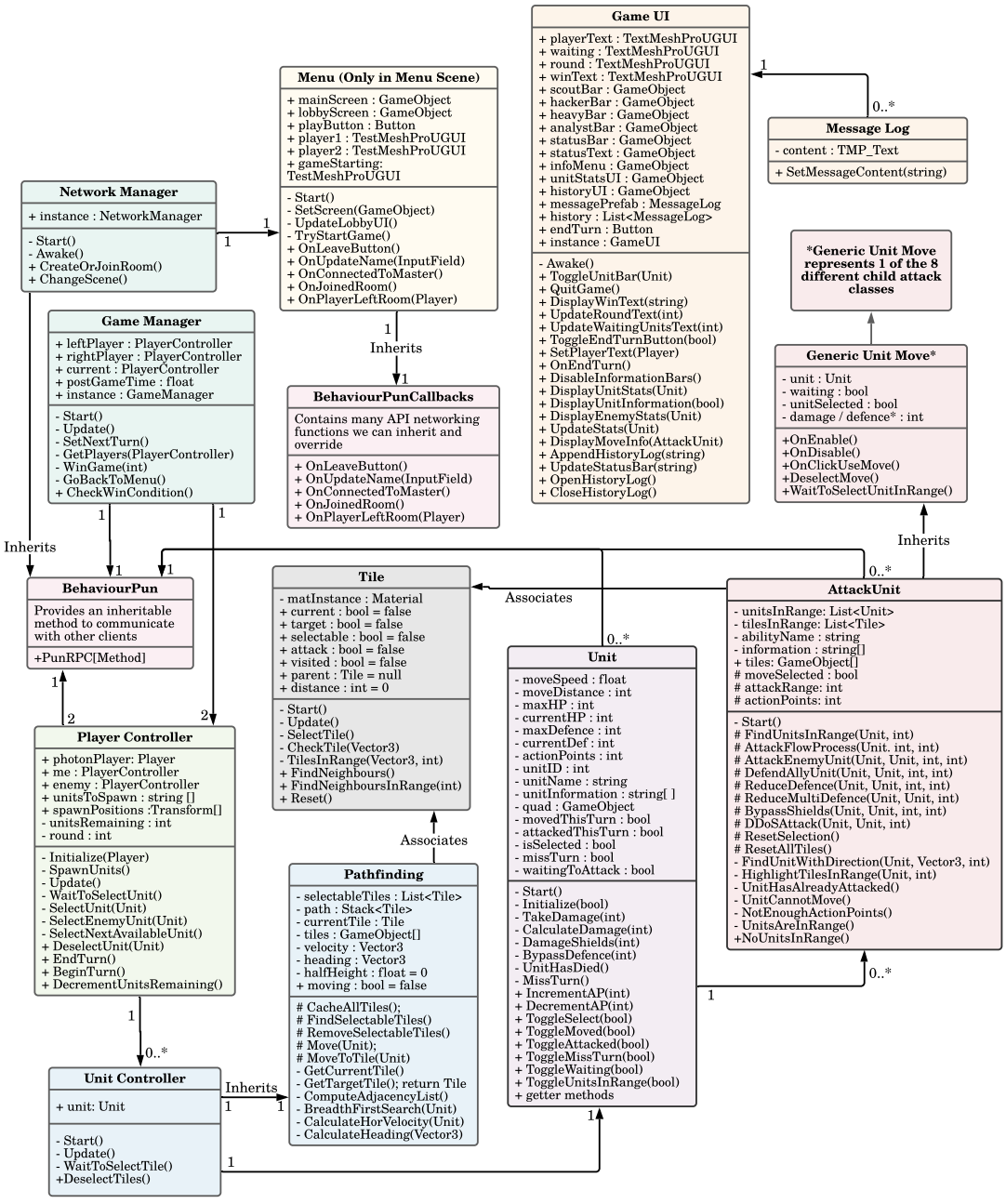


Figure 6: A Class Diagram Representation

4.4 Mapping the MDA Framework to Class Functions

Figure 7 highlights functions and components that specifically relate to the MDA framework. Due to the nature of Aesthetics being an emotional product of the game, in this diagram it represents all the physical assets and UI that the player interacts with.

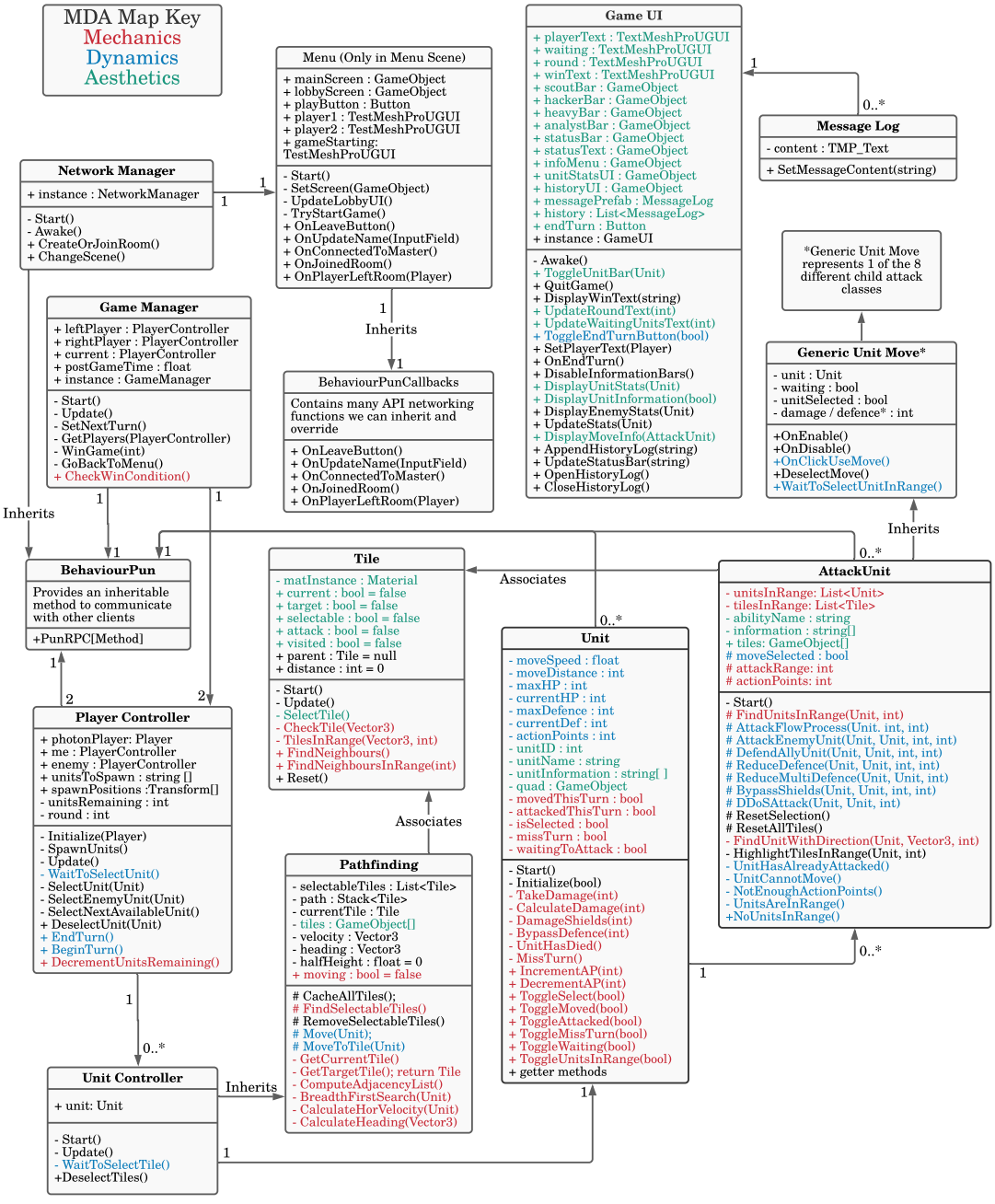


Figure 7: Mapping the MDA Framework to Class Functions

With reference to figure 6 and 7, an explanation of the core classes and components can be found below:

Game Manager
A singleton instance of this class forms the backbone of the project as it controls the game's flow sequence, turn-manager, and game scene.

Behaviour PUN Callbacks
This class is provided by the Photon Unity Networking 2 API and provides many networking functions that can be implemented and overridden.

Network Manager
A singleton instance of this class inherits the PUN2 API to handles hosting, joining, and leaving a room.

Menu
This is a relatively simple class that takes user input (from within the game menu) to invoke networking functions within the network manager.

Game UI
This script handles the transition and display of the user interface (UI) components.

Player Controller
A singleton instance of this class is instantiated for each player. It listens for player input once per frame. It allows the player to select units and manages the players-turn and status of all units each turn.

Unit Controller
Each Unit component has an instance of this class. It enables tile-selection and unit movement by inheriting functions provided from the Path-finding class.

Path-finding
This class utilises Breadth First Search to find the range of tiles that a selected unit can move to. It directly associates to each tile as it checks if it's unoccupied, and then highlights it green if it is a valid destination. The path-finding functions within this class were loosely adapted from a tutorial available at Game Programming Academy [17].

Unit
Each Unit component has an instance of this class. It contains unit information (e.g. health, defence, etc..) and functions relating to receiving damage, restoring defence and the unit's status (e.g. the unit has moved).

Attack Unit
Attack Unit provides functions for finding units in range, attacking, and defending units. It associates with Tile class to accomplish this (e.g. is a Unit located at this tile).

Generic Unit Move
Generic Unit Move represents the 8 unit abilities which are derived from the OWASP Top 10 (e.g. SQL Injection). These scripts inherit from Attack Unit for their shared functionality and are directly linked to the UI buttons which invoke the ability event upon user input.

4.5 UML Modelling - Activity Diagram

The following diagram refers to the sequence of two players joining a game, matchmaking and then the typical sequence of events that happens until the game is finished.

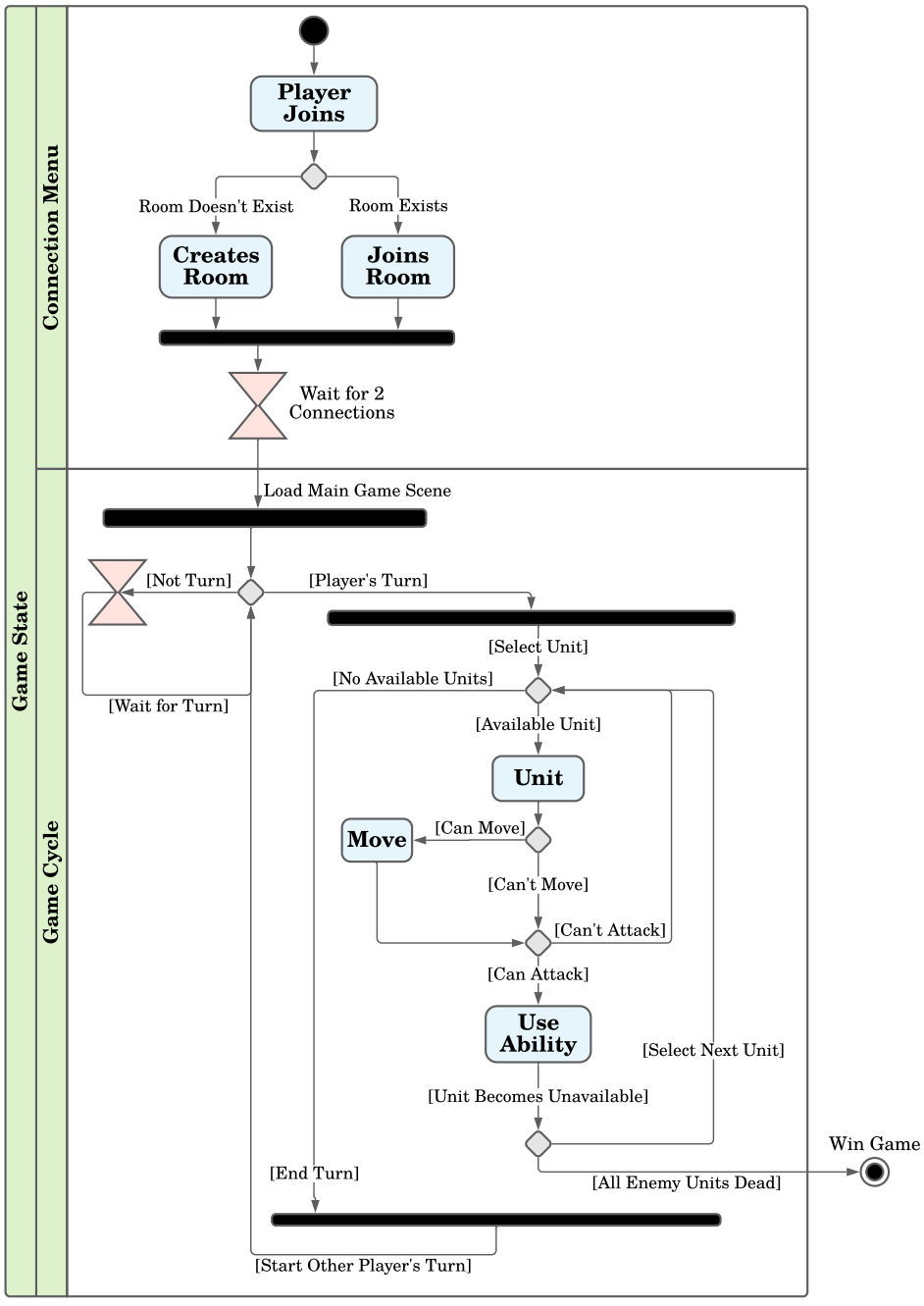


Figure 8: Activity Diagram of a Typical Match

4.6 Visual Design, Contrast and Accessibility

The main UI colour palette was inspired from NieR: Automata – a similarly themed title pivoted around security and machine intelligence. To design an application visually suitable for the web with appropriate complimentary and harmonic colours, this palette was entered into a colour checker [18].

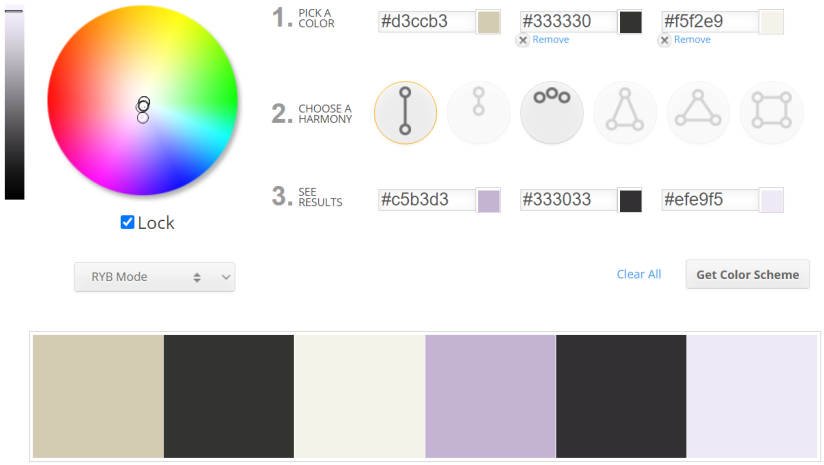
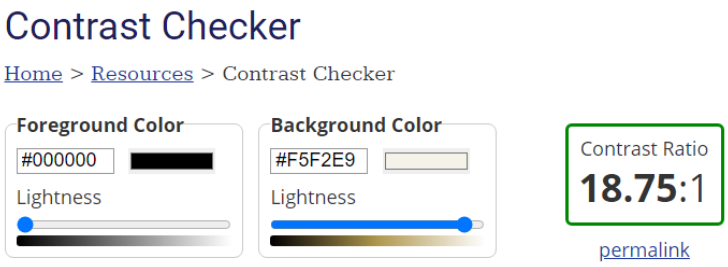
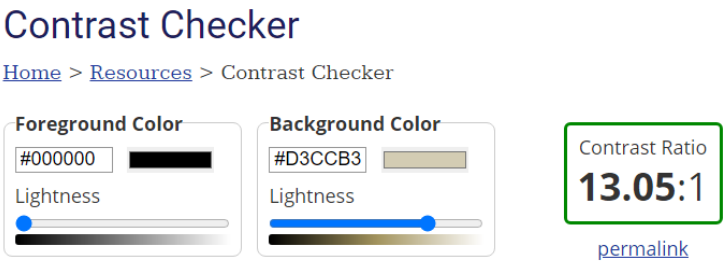


Figure 9: Identifying Complementary and Harmonic Colours

The Web Content Accessibility Guidelines (WCAG 2.1) requires that small text has a minimum contrast ratio of 7:1 [19]. All text used within the application has a contrast ratio of 18.75:1 and 13.05:1 which is compliant for all font-sizes.



(a) Larger Text [20]



(b) Smaller Text [21]

Figure 10: Contract Checking Text for Readability

4.7 Colour Blindness

Approximately 1 in 12 men (8%), and 1 in 200 women (0.5%), experience some form of Colour Vision Deficiency (CVD) [22] which exemplifies the importance of designing with accessibility in mind.

Figure 11 represents the initial design and choice of unit-colour before amending these changes with colour blindness in mind. As such, a colour-blind friendly palette [23], was implemented, alongside a royalty-free background pattern [24], which better contrasts the game-scene.

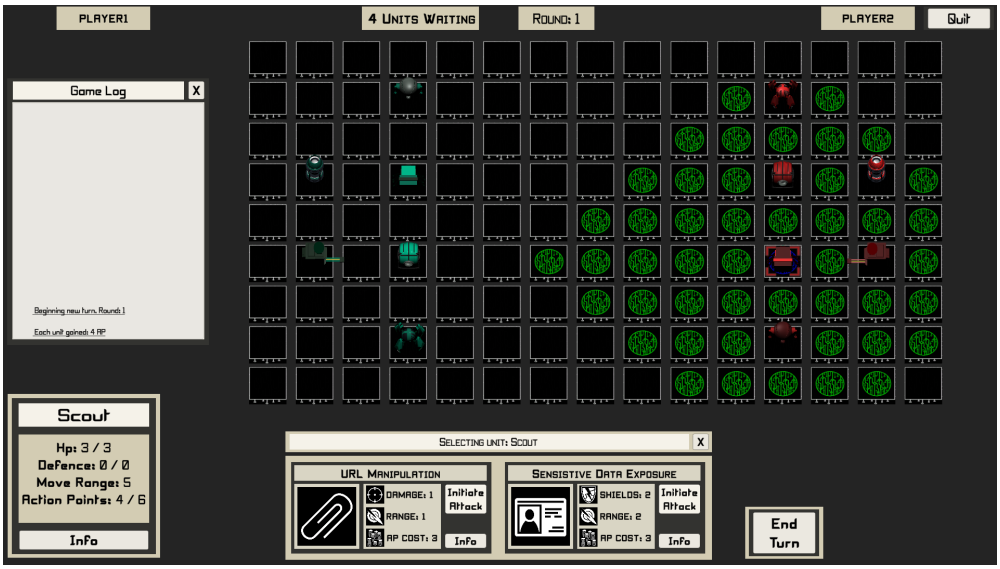


Figure 11: Initial Colour Scheme non-adhering to Accessibility Guidelines

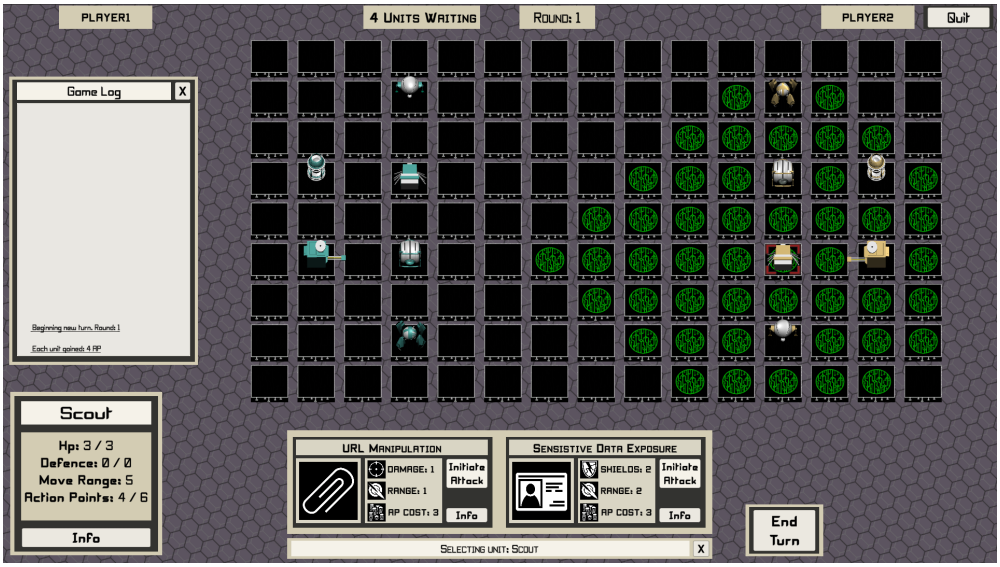


Figure 12: Revised Colour Scheme of the Desktop Application

Due to the technical limitations of WebGL compatibility for web browsers, the tile shader and animation failed to render. As such, an alternative simpler design was developed, as seen in figure 13.

To fulfil the requirements of developing cross-platform, the game was mapped such that two supporting UI resolutions were created (for a standard monitor aspect ratio of 16:9, and a tablet aspect ratio of 4:3). All buttons and game - interactivity is mapped with just the left mouse click (which enables touch screen support via tablet browser). The game does successfully play both in browser, and on desktop / mobile (cross-platform).

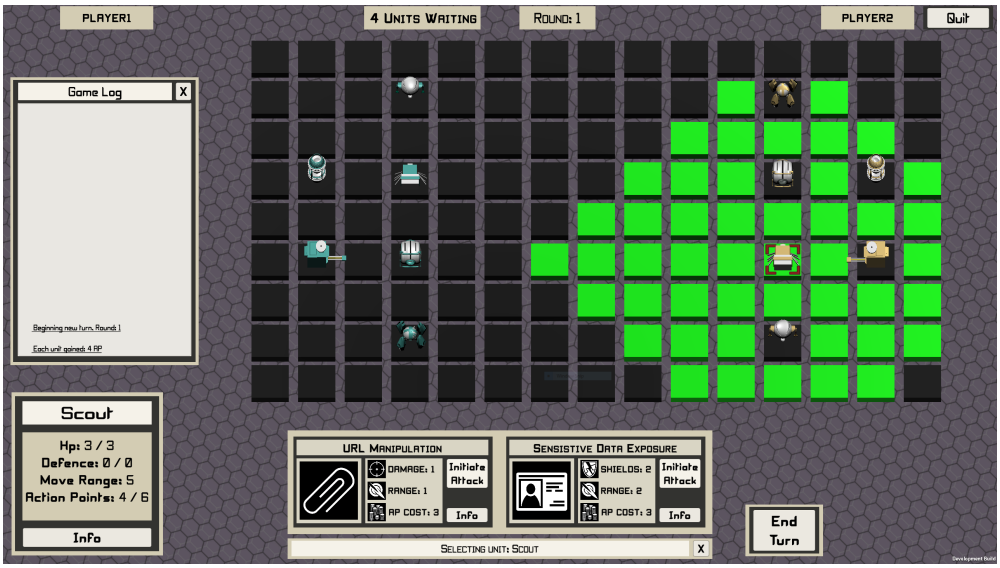


Figure 13: Revised Colour Scheme of the Web & Mobile Application

4.8 Design Patterns for Unity3D & C#

The following section exemplifies the recommended design patterns for Unity3D.

4.8.1 Component Design Pattern

This is the most common design pattern as Unity is a component-driven engine, in which game objects, scripts and functions are typically broken down into smaller sub-components [25].

The Unit game object is a good example of this as, all unit functionality (stats, movement control, networking operations and rigid bodies / mesh colliders), are broken down into sub-components and stitched together - figure 14. Similarly, to object-orientated programming, this approach enables components to be decoupled, re-arranged, and re-used easily.

A benefit of this approach is that unit's stats and abilities assigned (via the UI) can be modified and re-arranged at any time, allowing for flexibility of using one unit as a model for creating completely new units.

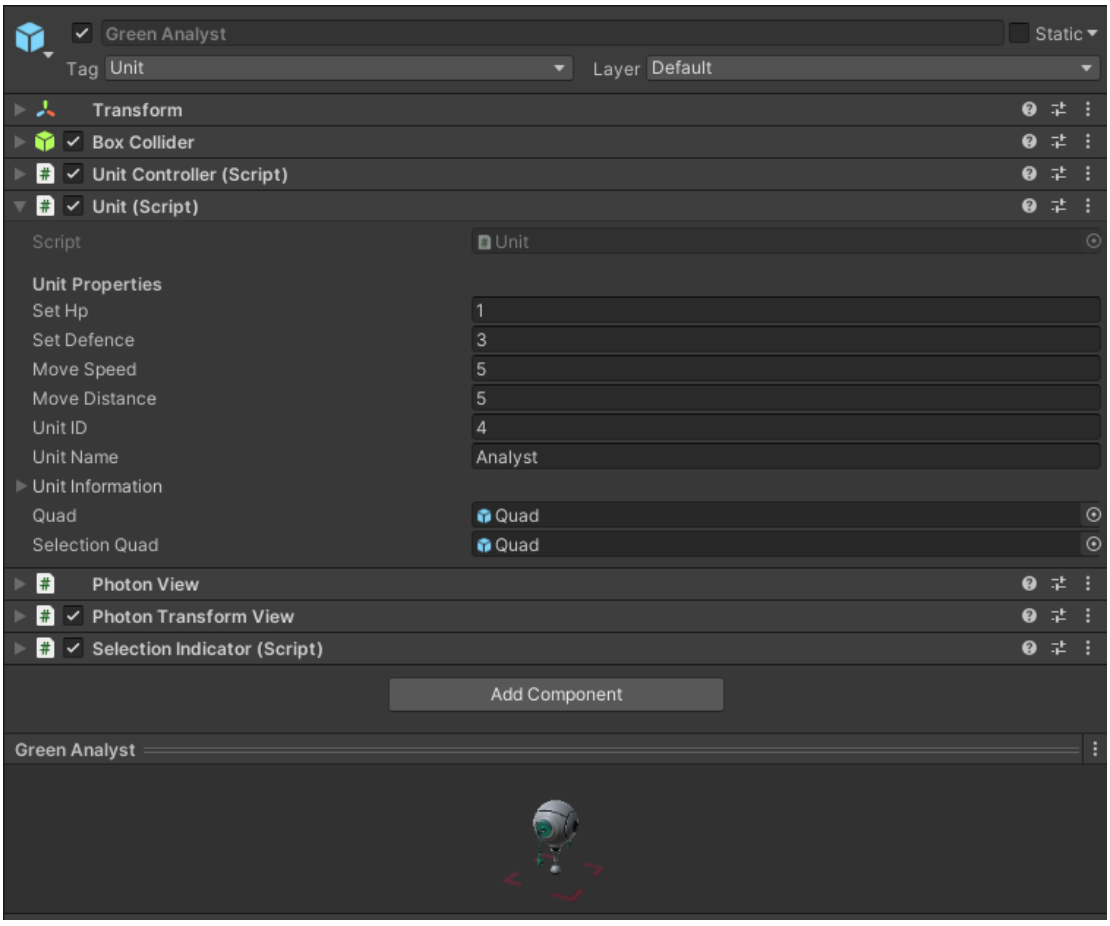


Figure 14: Example of Unit Components

4.8.2 Singleton Design Pattern

The singleton pattern involves creating a public static reference (of a non-static class) such that only one instance of that class exists [25]. This is incredibly useful for managers (e.g. the Game Manager) as it is not possible to instantiate static classes. However, having a single, static reference is versatile in that it stores important information about the state of the game – and can be publicly invoked by all other components without needing to worry of duplicated instances.

```
public class GameManager : MonoBehaviourPun {

    // Static singleton instance of the Game Manager
    public static GameManager instance;

    // Assign instance upon initialisation
    private void Awake() => instance = this;
}
```

Figure 15: Singleton Pattern

4.8.3 Flyweight Design Pattern

The flyweight pattern is designed to save memory by re-using instances of the same object [25]. For example, each time a tile changes colour, if a material instance is not saved and re-used, a new material object would be created which leads to unnecessary memory losses. This is important because as the size of the game scales (with hundreds of tiles), lag spikes will occur more frequently as the garbage collector must be invoked to remove old material instances.

```
private Material _matInstance;

// Cache material component on initialisation
private void Start() {
    _matInstance = GetComponent<Renderer>().material;
    _matInstance.color = Color.black;
}
// Re-use material instance each time the tile is updated
public void MarkSelectable() {
    selectable = true;
    _matInstance.color = Color.green;
}
```

Figure 16: Applying the Flyweight Pattern to Tile's Changing Material

5 Implementation

Unity3D [26] was chosen as an appropriate development environment because one of the key functional requirements identified was to design a game that could be exported to the web. Unity3D offers seamless integration with WebGL [27] to satisfy this requirement.

Phaser3 [28], HTML and NodeJS were considered as an alternative, web-development based solution. However, Phaser3 is still a relatively small open source project whereas Unity3D is a much more widely used industry trade skill. Furthermore, Unity3D has the largest online community, ample resources and integration for supporting frameworks.

5.1 Client - Server Multiplayer

Photon Cloud is a software as a service (SaaS) solution which provide, and host, cloud servers for free up until 20 concurrent players [29]. Beyond this, Photon offer paid services for more scalable solutions. Furthermore, Photon provides a free-to-use API, (Photon Unity Networking 2 [30]) which enable developers to integrate multiplayer functionality into their games. PUN2 was chosen because it has an ample tutorial base and very strong integration with Unity3D. Lastly, PUN2 supports cross platform gameplay which fulfils the requirement of being able to export and play on the web.

Mirror [31] is an alternative free and open source library which was built to replace Unity's previous deprecated multiplayer libraries (UNET). Although Mirror is completely free to use, relatively simple and highly covered, Mirror does not provide any functionality for externally hosting a game session. This would require developing, and running, a server lobby constantly to allow people to connect to the game. Clearly, this is would not be a viable long-term solution.

5.2 Photon Unity Networking 2 Architecture

PUN2 follows a classic Server – Client architecture. When a client connects to the Photon Network initially, it invokes the 'JoinOrCreateRoom()' function which essentially tries to join a non-full room (if it exists), or create one if there are no other rooms (with space) available.

The first player (client) to do this becomes the master client which is responsible for hosting the game. If the master client disconnects for any reason, there are options for the next available client (if present) to be promoted to a new master client. However, since this project is only designed for two players, it returns the other player to the main menu and ends the room session.

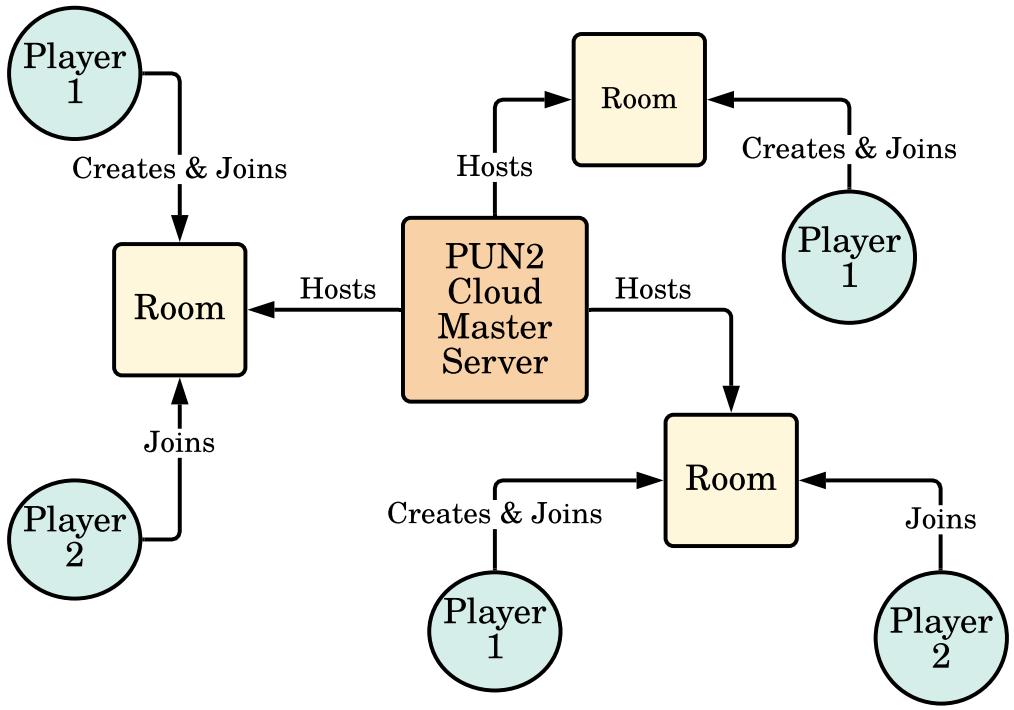


Figure 17: PUN2 Architectural Design

5.3 Remote Procedure Calls - RPCs

PUN RPCs (provided by the PUN2 API [32]) enable client - client communication in which method functions can be invoked on the opponents game version remotely. As both clients are running different, independent images of the game, RPCs are necessary to match these images and important state variables.

For example, after a unit has attacked another unit, it will remotely call the "Take Damage" function which PUN will invoke on the recipient client.

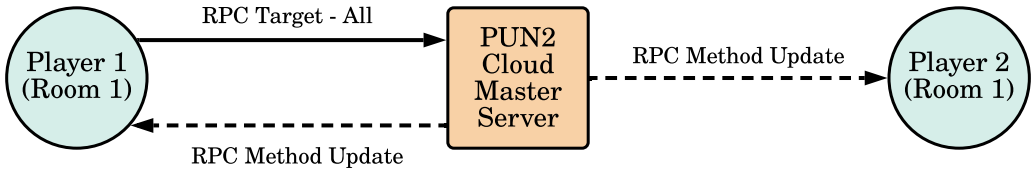


Figure 18: RPC Communication to One User

By calling 'RPC Target.All' however, the function is executed on all clients connected in the room (including the place of origin). This is the most often case since the majority of multiplayer functions are state-changing (such as beginning a new turn, and editing a units states), and these values must remain concurrent.

Figure 19: RPC Communication to All Users

5.4 Implementation of Connection Menu

The following auto-match makes (for simplicity) which fullfils requirements: [].

Figure 20: Welcome Menu

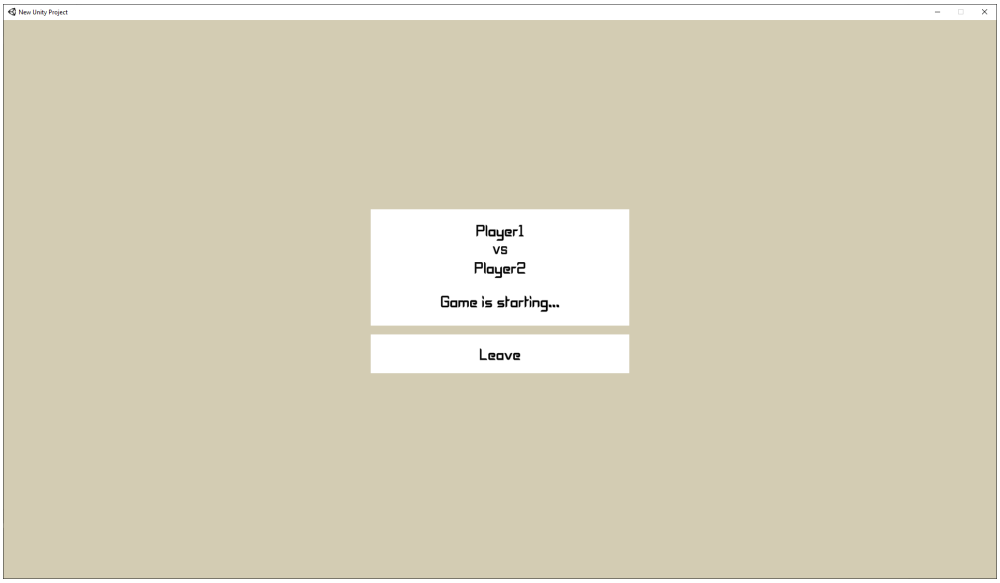


Figure 21: Connection Screen

5.5 Implementation of Movement

Movement parameters can be modified within the unit's inspector - and freely changed for any unit. This fullfils requirement. Figure 22 illustrates the range of tiles a selected Scout Unit can move to.

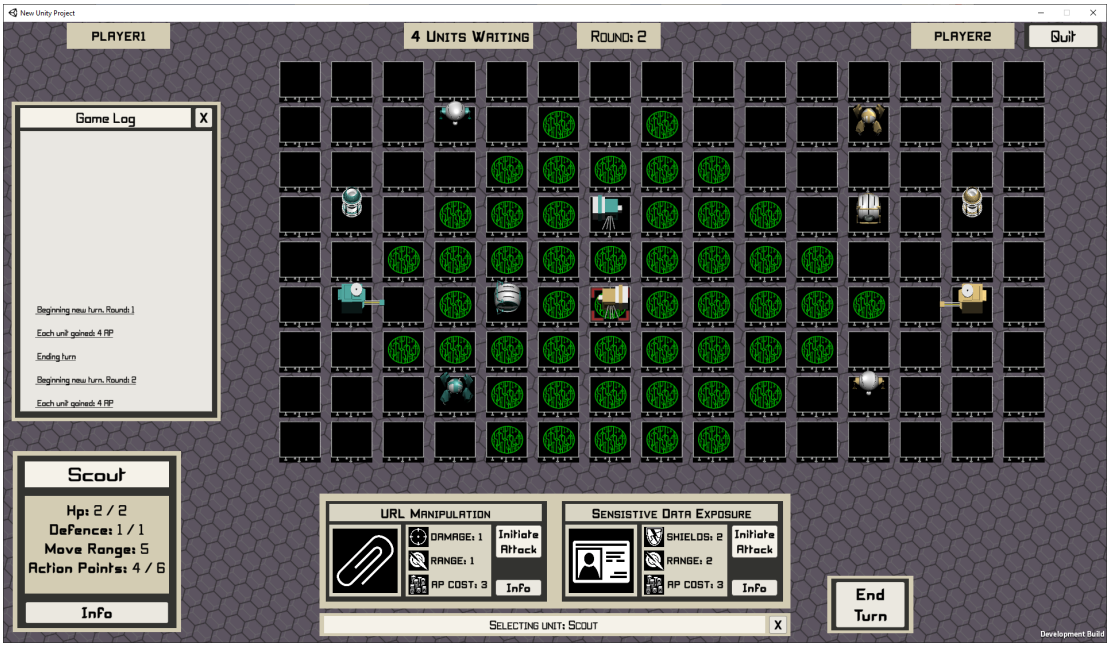


Figure 22: Movement Options for a Scout - with Range 5

5.6 Movement Mechanics - Breadth First Search

To calculate the selectable tiles in range of a unit, Breadth First Search (BFS) was implemented as it always returns the shortest path to the destination.

When a unit is selected, a ray-cast is sent below to find the tile underneath. From this tile, four more ray-casts are emitted north, east, south, and west to discover any neighbours. If these neighbours are walk-able, they are added to adjacency list attached to that tile. Occupied tiles are ignored.

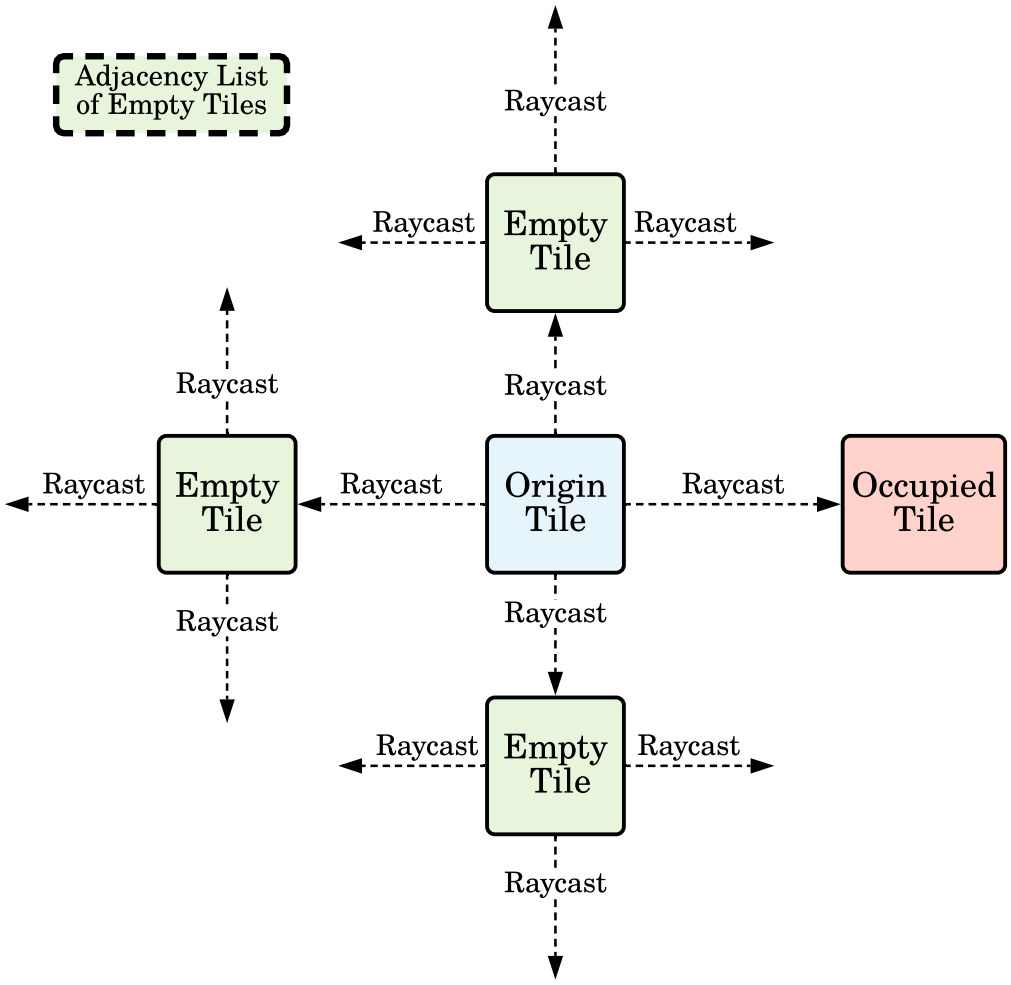


Figure 23: Ray-casting to Adjacent Tiles

BFS utilises a queue data structure. The starting tile is enqueued first. This is then dequeued such that it is marked as selectable, and all adjacent (walk-able) tiles can be processed. Each tile is marked as visited so that these tiles are not checked twice. See Figure 24 for a diagrammatic representation, and figure 26 for a programming explanation.

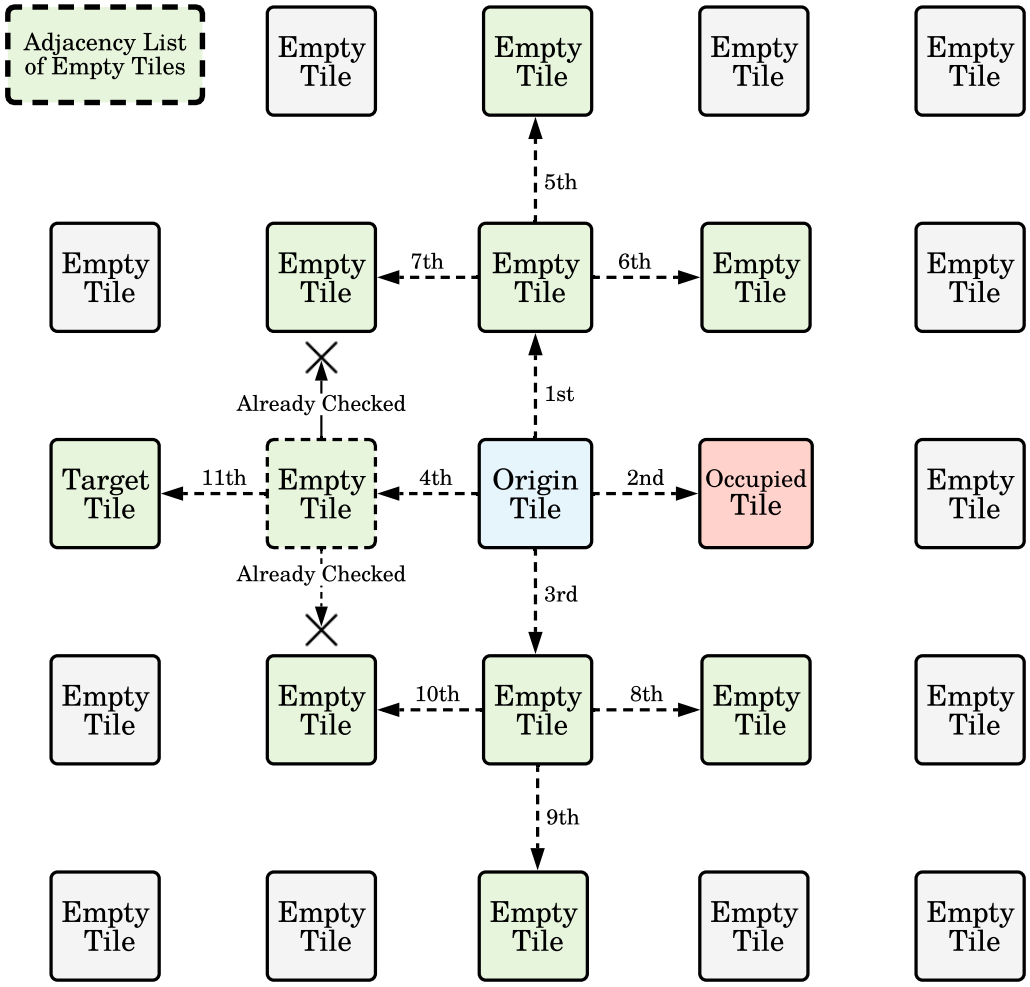


Figure 24: Finding Selectable Tiles via Breadth First Search

For each adjacent tile that is processed, the parent tile is assigned to it. When a target tile is selected, it's parent is recursively accessed until we arrive back at the origin tile (see figure 25). As such, this always returns the shortest path to the destination.

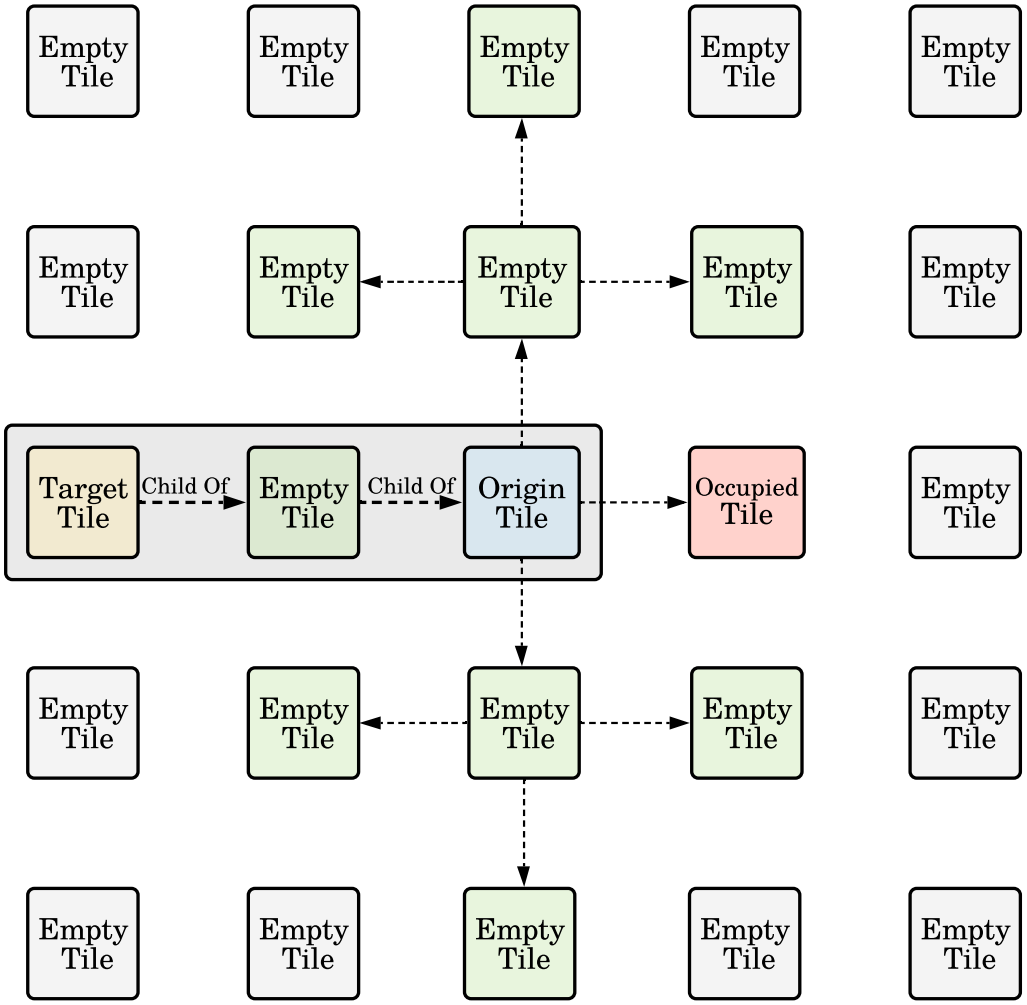


Figure 25: Finding the Shortest Path to Target Tile

Figure 26 demonstrates the core implementation of breadth first search.

```
        private void BreadthFirstSearch(Unit unit) {

        Queue<Tile> tiles = new Queue<Tile>();
        // Enqueue the tile underneath selected unit
        tiles.Enqueue(currentTile);
        // Mark as visited so this tile isn't processed twice
        currentTile.visited = true;

        // Begin dequeuing queue, whilst enqueuing adjacent
           tiles
        while (tiles.Count > 0) {

            // Dequeue the first tile and mark as selectable
            Tile tile = tiles.Dequeue();
            selectableTiles.Add(tile);
            // Highlight the tile green
            tile.MarkSelectable();
            // If the tile is outside of range, continue to
               the next tile in the queue
            if (tile.distance >= unit.GetMovementDistance())
               continue;

            // Search through the 4 adjacent tiles
            foreach (Tile t in tile.adjacencyList) {

                // If tile has been visited, continue to the
                   next tile in the adjacency list
                if (t.visited) continue;
                // Assign the parent of the adjacent tile
                t.parent = tile;
                t.visited = true;
                // Update distance of tile from unit to keep
                   track of how far we are from the starting
                   tile
                t.distance = 1 + tile.distance;
                // Enqueue tile from adjacency list
                tiles.Enqueue(t);
            }
        }
    }
```

Figure 26: Implementing Breadth First Search

5.7 Implementation of Ability Range

Ability range's work similar to pathfinding, however they don't utilise breadth first search, and instead opt in a "line-of-sight" option. Again this move can be scaled for any positive integer. This satisfies requirement [ ]

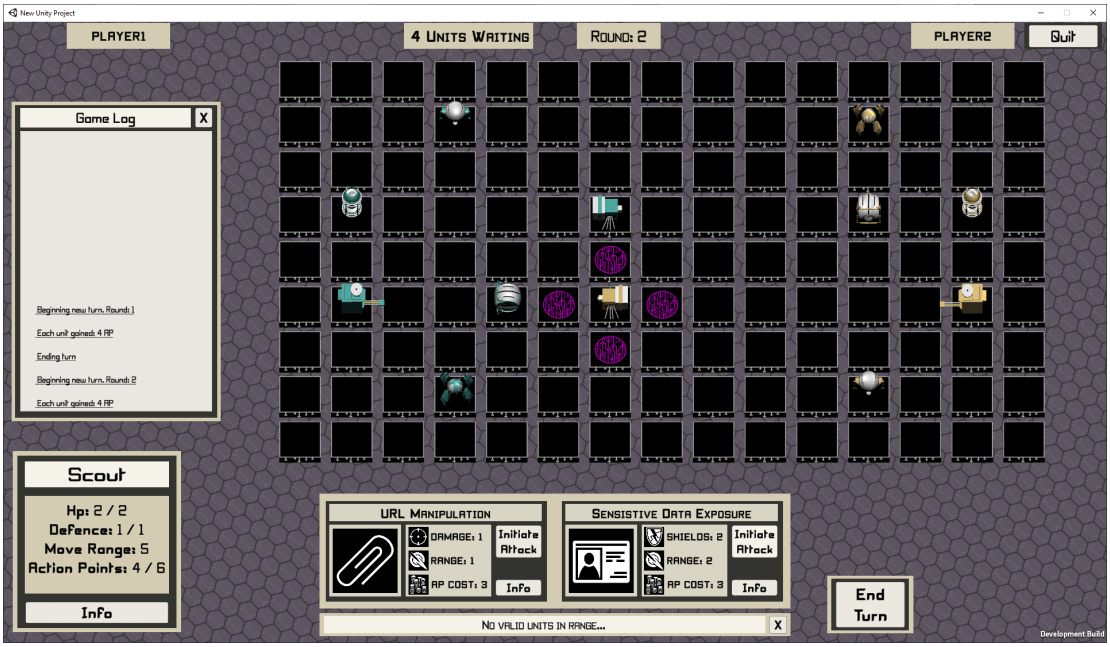


Figure 27: An Ability with 1 Range

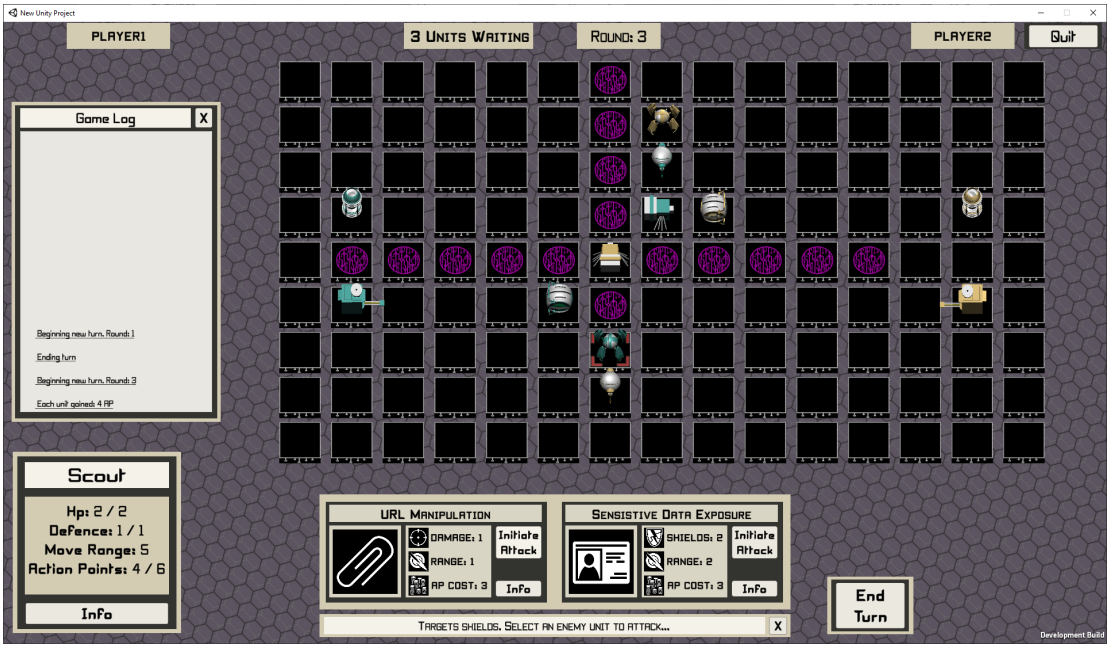


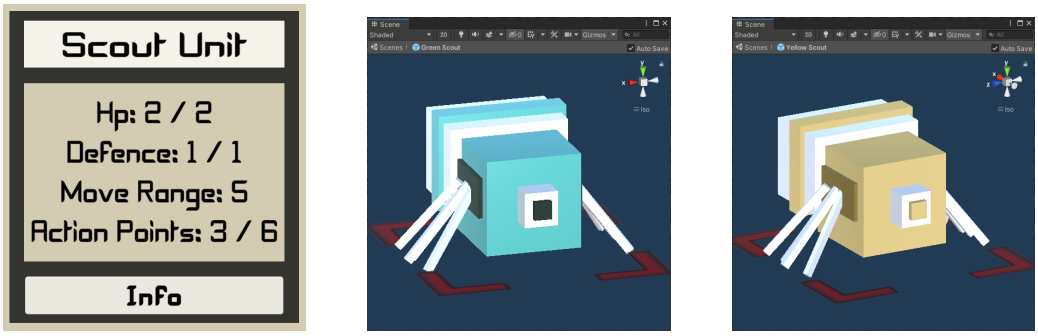
Figure 28: An Ability with 5 Range

5.8 Implementation of Units

The following sections describe the implementation of selectable units and their following abilities which are derived from the OWASP Top 10 2017 [16]. All models are royalty - free and accessible on Blend-swap. All units have a maximum of 6 action points (and can gain up to 4 per turn).

5.8.1 Implementation of Scout Unit

The scout unit represents exploiting broken control and exposing sensitive information from poorly designed web applications. It has access to the following moves: URL Manipulation and Sensitive Data Exposure (figure 30).



(a) Scout Stats (b) Green Scout (c) Yellow Scout

Figure 29: Scout Unit Stats and Model [33]

URL Manipulation typically involves changing the URL parameters in order to bypass authorisation control. As such, the core ability of this move is to bypass the enemy unit's defence. This makes it a good move against units with high defence, but low HP (such as the front-end web server). This move is close range as it requires the unit to directly interact with the web application.

Sensitive Data Exposure includes leaking personal, account and financial information. Improper handling of sensitive data in transit or storage lead to data breaches which can cripple a company's reputation. In more severe outcomes, such information could be used to commit fraud and identity theft. As such, this move is a highly damaging defence-reducing attack.

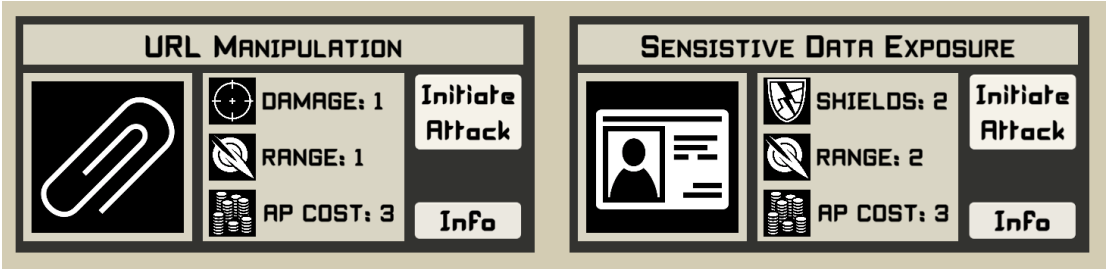
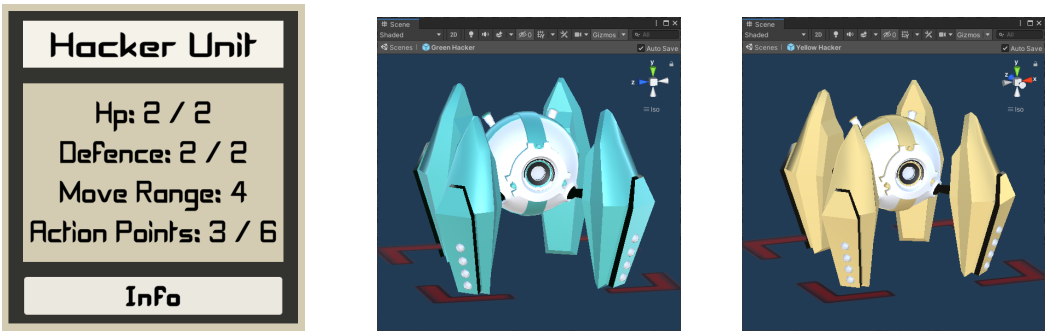


Figure 30: Scout Abilities - URL Manipulation and Sensitive Data Exposure

5.8.2 Implementation of Hacker Unit

The hacker unit specialises with direct attack vectors typical of penetration testers who can prod a web application in search of vulnerabilities. SQL injection and XSS scripting are two very common methods that can be brute-forced (figure 32).



(a) Hacker Stats (b) Green Hacker (c) Yellow Hacker

Figure 31: Hacker Unit Stats and Model [34]

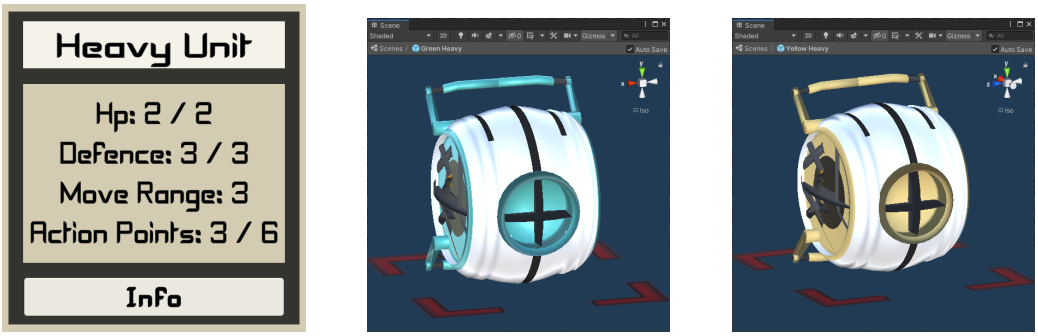
Cross-Site Scripting (XSS) attacks can occur when a web application fails to validate user input by escaping special characters. As a result, an attacker can execute scripts in a victim's browser, hijack their session data and even redirect them to other phishing sites. Therefore, this move is a highly damaging close range move which does +1 critical damage to the web server.

Sequel Query Language (SQL) injection exploits the interactivity between the web application and commands that query the back-end database. Using SQL injection, it is possible to bypass authorisation, extract sensitive information from databases and even manipulate database entries with your own custom information. Therefore, SQL Injection is a longer range move which does +1 critical damage to the database server.

Figure 32: Hacker Abilities - SQL Injection and Cross-Site Scripting Attack

5.8.3 Implementation of Heavy Unit

The Heavy unit represents physical disruption by disabling operations externally and internally. It has access to the following moves: Distributed Denial of Service and Delete Security Logs (figure 34).



(a) Scout Stats (b) Green Heavy (c) Yellow Heavy

Figure 33: Heavy Unit Stats and Model [35]

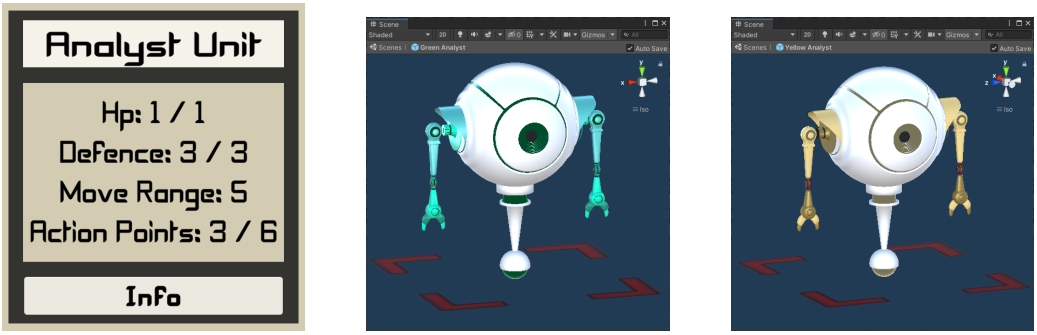
A **Distributed Denial of Service (DDoS)** involves using a network of devices in order to flood a web application, network or server with overwhelming internet traffic - and disable it from being able to handle requests. This move will disable any movable unit for 1 turn. If used on the web/database server, it will halt action point for all units for one turn.

Delete Security Logs exploits Insufficient Logging & Monitoring (figure 5). Insufficient data logging and monitoring allows an attacker to persist in a compromised system without being detected. Logging pertains to recording anomalies, failed login attempts, As such, this move is a multi-targeting hitting move (within attack range), that lowers the defence of all of those units.

Figure 34: Heavy Abilities - DDoS Attack and Delete Security Logs

5.8.4 Implementation of Analyst Unit

The analyst is designed to represent the role of security analysts who specialise in defending a network. This includes implementing security firewalls and keeping systems up to date. With regards to interactivity with the web application, the analyst is also responsible for monitoring and logging internet traffic in order to detect, and react, to malicious activity.



(a) Scout Stats (b) Green Analyst (c) Yellow Analyst

Figure 35: Analyst Unit Stats and Model [36]

A **Firewall** is required to filter unwanted traffic and keep web-application components safely isolated. Firewalls can be used in conjunction with an Intrusion Detection/Prevention system in order to detect malicious activity and send alerts upon detection. As such, this move restores the shields of one unit in range.

Implement Access Control is designed to mitigate Broken Access Control (figure 5). A good access control system should be implemented once, and periodically reviewed. Control policies based on user roles should enforce the least amount of privileges each user requires to complete their interaction within the web application. As such, this move restores the defence of all units range.

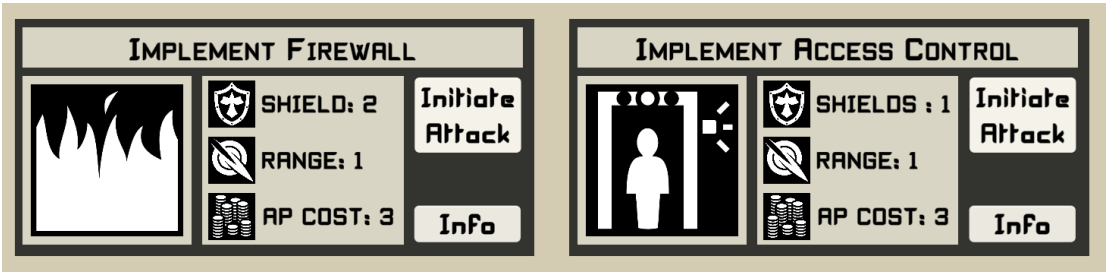
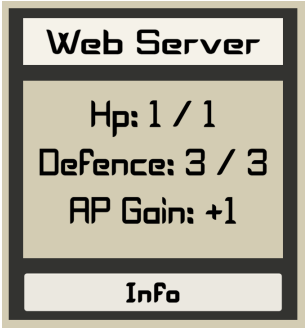


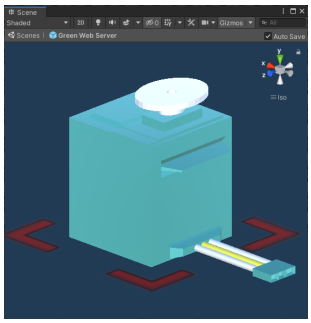
Figure 36: Analyst Abilities - Implement Firewall and Access Control

5.8.5 Implementation of Server Units

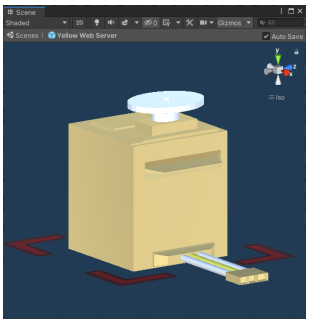
The Web Server is designed to represent []



(a) Web Server Stats



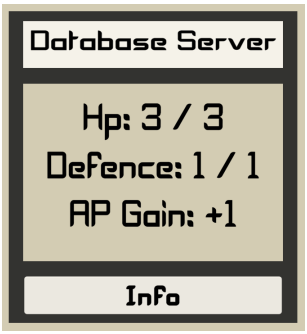
(b) Green Web Server



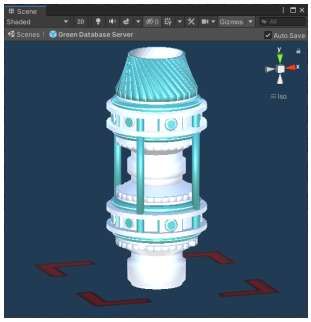
(c) Yellow Web Server

Figure 37: Web Server Stats and Model [37]

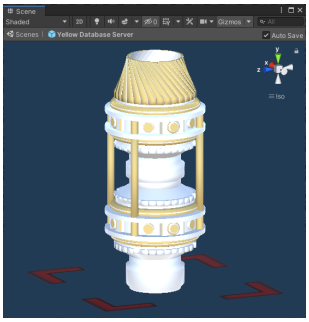
The Database server is designed to represent [ ]



(a) Database Server Stats



(b) Green Database Server



(c) Yellow Database Server

Figure 38: Database Server Stats and Model [38]

Both these server objects generate +1 action points per turn to all other units, unless disabled or destroyed.

5.9 Implementation of Information Menu

Each unit, or ability, has a "info" button which can be clicked to invoke more information about the attack nature, and in which situations it's recommended to be used. This fulfils user story requirement [Ref - information] and partially [Tutorial]].

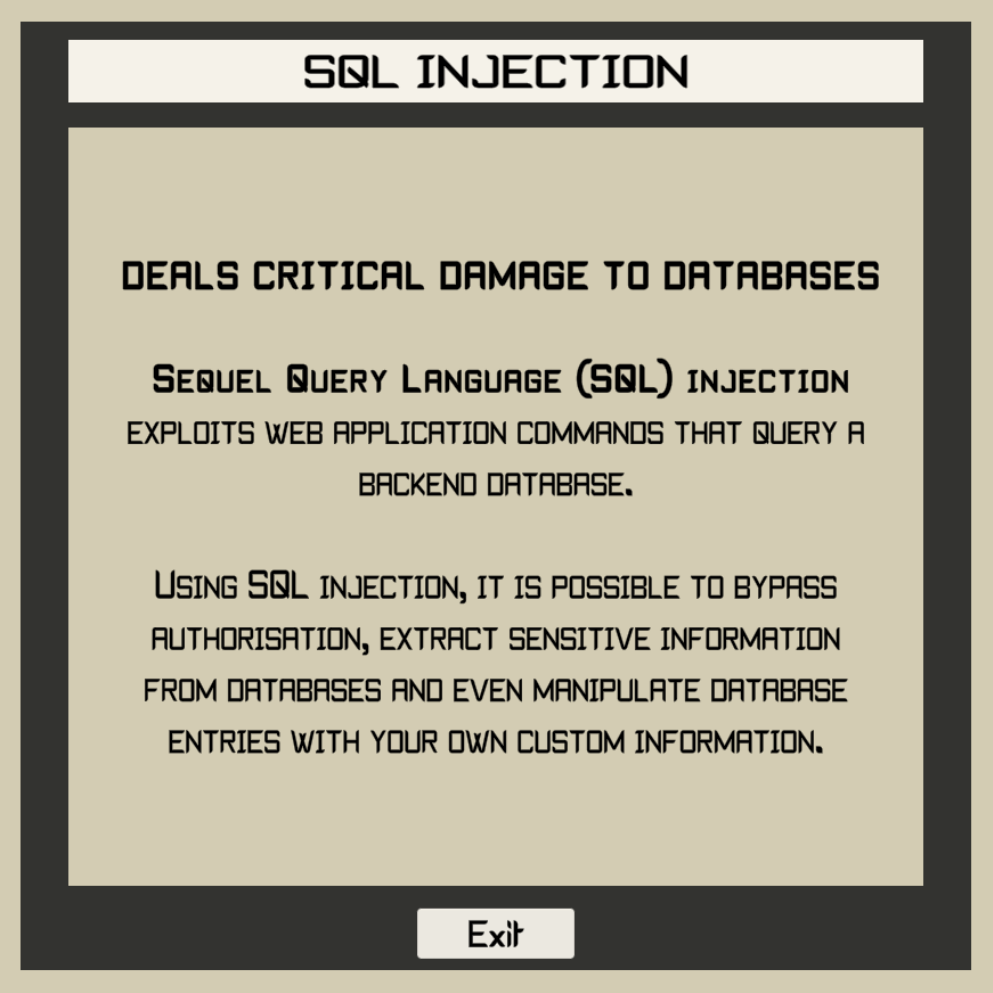


Figure 39: Information Screen Example - SQL Injection

5.10 Implementation of Status Bar

The tutorial also partially instructs the player what commands to use as it was not possible to develop a fully blown tutorial. As such, this partially fulfils user requirement (tutorial).

Figure 40: Status Bar Example - Using Delete Security Logs

5.11 Implementation of History Log

So the player can keep up and follow the actions of themselves and the enemy player, each unique ability and game interaction is recorded in the history log - figure 41. The history log fulfils user story requirement (understanding).

Figure 41: History Log

6 Testing and Evaluation

This section explores the testing strategies implemented - which primarily include live debugging (with ParallelSync), unit testing, identifying edges cases and performance analysis (with the Unity profiler).

Alongside Unity, JetBrains Rider was utilised as an integrated development environment for its inbuilt functionality of providing performance recommendations and highlighting programming flaws.

6.1 Unit Testing

For small single-player games, play-testing in Unity3D is typically very easy via simply running the game. However, as a project expands, it can be a very time-consuming process to launch the game, load menus, connect to another player, and perform actions to reach a desired game state to check one thing. As such, automated testing on pure functions is a necessity.

The following unit tests were performed on all pure functions of the game that should always return the same output on a given input. These include using unique moves with a range of parameters, changing the state of selectable units, and the functional correctness of all calculations (such as calculating damage and action points).

Figure 42: Unity Test Runner

6.2 Identifying Edge Cases

Whilst testing these functions, all edge cases were identified and tested (such as negative, positive and zero values). A large benefit of test-driven development is that it enforces good programming practises in that functions only perform one operation, can be debugged easily and flexibly re-written (without breaking other components). A list of these tests can be seen in Appendix A.

6.3 ParallelSync for Live Builds

ParallelSync [39], is a Unity open source extension which clones the Unity development environment. This is fundamentally important for play-testing a multi-client game as you cannot have duplications of the same environment open concurrently. As a result, this would require rebuilding the project externally which can take several minutes each time (especially for WebGL development builds).

ParallelSync works by cloning the folder structure of the project without needing to duplicate all files, assets, and game scenes. Instead it establishes pointers to the original file location with read only access. Because of this, it is safe and easy to create, load and destroy as many clone environments as required.

6.4 Performance Testing

Unity's integrated profiler was utilised to analyse the performance of the application. Since this is a time-consuming process, it is recommended to complete this at the end of a significant feature implementation [40]. Profiling was done comparatively in that sections of the game were recorded, saved, and analytically compared to its previous performance. It is wise to profile after playing a game for an extended amount of time such that application has reached a sustained state.

Unity profiling is an instrumentation-based profiler [40]. At each stage, a 'snippet' of 300 frames of is captured by inserting a small marker in every function call – and calculating the execution time of this function. Although this creates a tiny bit of overhead, the overall profile is unchanged. Figure 43 demonstrates the overall frame rate captured – which sustained a mean average of 6.97ms (143 frames per second). Although this fulfils requirement [X], there were occasional lag spikes which reached up to 28.44ms (35 frames per second).

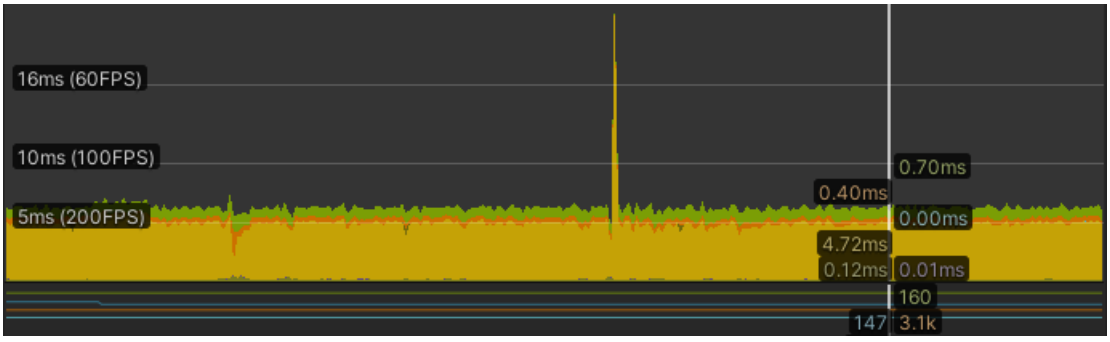


Figure 43: Average FPS of a 300-frame sample

6.5 Achieving a Target Frame Rate

In each frame, the core update loop is processed by the CPU and will require a certain number of milliseconds to complete; this time is inversely proportional to the frame rate [41]. Therefore, to achieve a target frame rate, the core update loop must be less than, or equal to, a target budget which is given by:

$$Budget = 1000\ ms\ /\ Target\ Frame\ Rate$$

To achieve 60 FPS, the target budget would be equal to: $1000\ ms\ /\ 60 = 16\ ms$. Therefore, the application's target budget is 16ms - such that each frame within the player's update loop is executed in 16ms or less consistently.

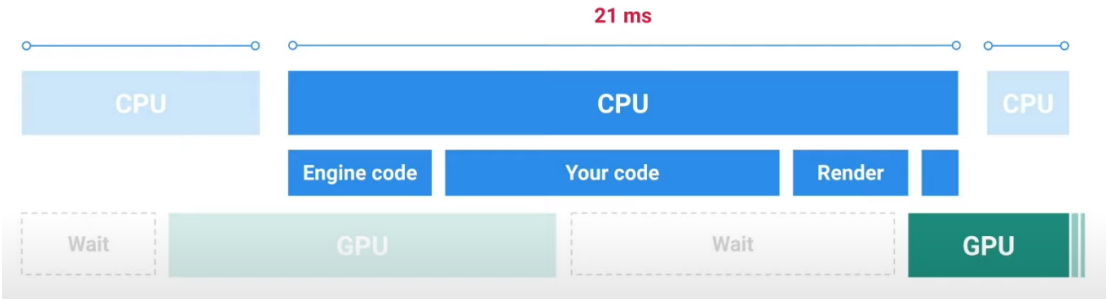


Figure 44: Structure of a Frame [41]

6.6 Evaluation and Optimisation

The first notable observation was investing the performance of each tile updating its status per frame. Since the application scenario had 135 tiles, this meant there were 135 unnecessary tile updates per frame (occupying 15.13% of the players update loop – Appendix 2.1, figure 48. Although this only occupies 0.15ms (0.9% of our target budget), user requirement [X] – scalability was to create an application that could be scaled up in size and scenarios. If this game was to become open world with 1000+ tiles, this would become 9% of our target budget. To amend this, the code was reworked such that tiles no longer updated their status manually, and we instead coupled to other objects invoking a tile status change when necessary. The result can be seen in Appendix 2.2, figure 49.

From profiling, it was discovered that the garbage collector was being invoked 544 times each frame to re-allocate 20kb of memory (figure 45 and figure 46. This stemmed from breadth first search path-finding calculations being run every single frame, initiating 493 ray-casts to adjacent tiles. This was significant as the number of ray-casts (incurred by the Breadth First Search function) scales with the number of tiles, and at 135 tiles – was already occupying 0.89ms (5.3%) of our budget.

After discovering and amending this such that path-finding calculations were only ever invoked once (upon selecting a unit), there were no evident garbage collector

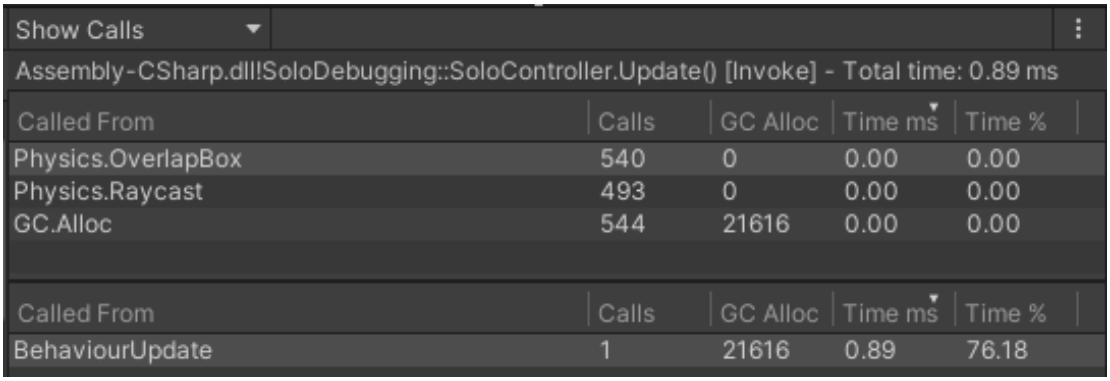


Figure 45: 544 GC Allocations per Frame (21kb)

calls within a 300 frame – and more importantly, unit's that were 'shaking' within the WebGL version of the game moved a lot more fluently.

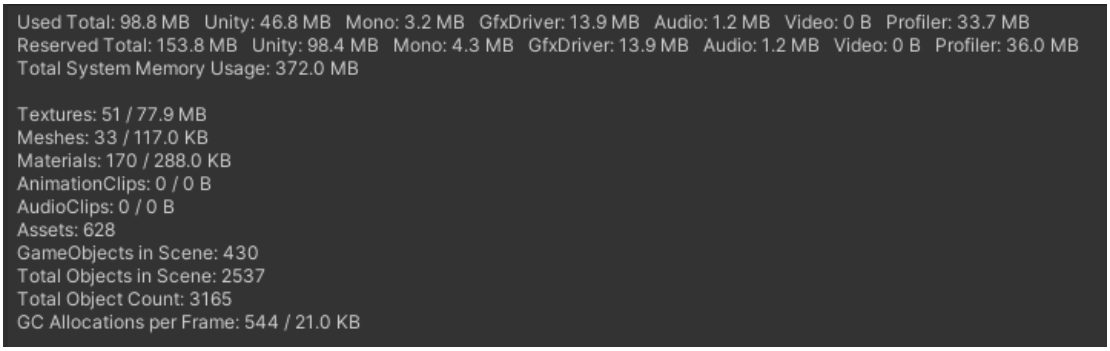


Figure 46: Memory Usage (per 300 frame capture)

To conclude, after optimising the game, the mean frame budget was lowered to 6.97 (1.69% less per frame), with no notable lag spikes exceeding the budget of 16ms. This can be seen Appendix 2.1 51,

6.7 Evaluation of Project Completion

Table 5 illustrates the completion status of all user stories established from the functional and non-functional requirements.

ID	functionality or feature	MoSCoW	Completion
1	View a unit's stats (HP/Defence)	Must	Yes
2	View an abilitiy's range, damage & cost	Must	Yes
3	View a colour-blind friendly UI	Should	Yes
4	View a unit's movement range	Must	Yes
5	View a unit's ability attack range	Must	Yes
6	Move a unit	Must	Yes
7	Attack a target	Must	Yes
8	Disable the target	Must	Yes
9	Defend an ally unit	Must	Yes
10	Click on a move description	Must	Yes
11	Click on a unit	Must	Yes
12	Click on and view a history log	Should	Yes
13	View the units remaining	Should	Yes
14	Pan the camera and/or zoom in	Could	No
15	Unique animations for each ability	Could	No
16	Varied unit abilities	Could	Yes
17	Cost weighting for each ability	Should	Yes
18	An intermediate tutorial	Should	Partial
19	A help / status bar	Should	Yes
20	Multiplayer functionality	Must	Yes
21	Single player functionality	Won't	N/A
22	Quit / leave the session	Must	Yes
23	Win or lose	Must	Yes
24	Play on Desktop	Must	Yes
25	Play on Web Browser	Should	Yes
26	Play on Tablet	Could	Yes
27	Play cross-platform sessions	Could	Yes
28	Learn about the OWASP #10	Must	Maybe
29	Purchase & upgrade new units	Could	No
30	Receive feedback after a game	Could	No
31	Hear gameplay music	Could	Yes

Table 5: Completion Status of User Stories

The core goal of the project was successful in that all 'musts' were completed with

exception of user story 28.

User story 28 refers to how effective the gamification mechanics were implemented at teaching the OWASP Top Ten which is impossible to evaluate without user feedback. However, due to the time constraint of this project, receiving such feedback was not possible. The Design-Dynamics-Experience framework [42], exemplifies this as it builds on from the flaws of the MDA framework by evaluating the user experience from the overall design and existing dynamics. This approach however understandably takes more time as it adds more components into the overall development process.

Furthermore, all shoulds (bar the Tutorial) were fully implemented, with shoulds having a partial completion since there are existing features (status and history log and information / help menu) which can be used to understand the core gameplay. The game was also built in a simple point and click way (where every action is mapped to the left mouse click - in order to support mobile touch screen support too). However, a full tutorial implementation was set last in the development cycle since it's core necessity would be to enable receiving feedback and evaluation which also wasn't possible.

Unfortunately, although not all the 'coulds' were completed, however with regards to user story 29 and 30, these weren't necessarily the most important. User story 30 was identified as a very important gamification mechanic, however since the nature of this game is similar to chess (with very strongly thought out feedback upon game operations), it would be far beyond my skill-set to develop a similar sort of automated/evaluation feedback system. User story 28 also was built upon the idea of developing a game that can be expanded for future projects - which was satisfied by the nature of which the unit "framework" is incredibly modular and can easily be expanded upon (by allowing them to unlock new ability moves and so on).

6.8 Evaluation of Approach

Using Unity3D, paired with Photon Cloud and Photon's own networking API proved to be a really good implementation of the project. Photon's own API has ample documentation, and developing in Unity3D made exporting game styles, debugging and testing for both windows and WebGL builds incredibly easy. As such it was possible to develop a cross-platform game that's both playable on multiple devices (PC, Web and Mobile), as well as cross-playable between these devices. As such this meets the biggest requirement identified in the literature review - similar to A Game of Threats, forming a cross-platform game that could be played in a workplace environment (such as trainining environments, breaks and sessions).

Since the game was developed with mobile controls in mind, everything was mapped to the left click which can be emulated and played successful with touch screen controls. The overall frame-rate and interactivity on a mobile device performs as

well as it does on a desktop, point-and-click web browser.

6.9 Evaluation of Networking Strategy

Photon Cloud and it's networking API (PUN2) proved to be an incredibly useful and efficient tool for creating a cross-platform multiplayer game. Not only does it not care for the version of the game that's being played on, there was also almost no perceivable impact to performance or frame-rate when profiling the game in single player (to evaluate efficiency of implemented mechanics) vs having two multiplayer clients. This networking API was also incredibly easy to interpret the documentation, and override important methods for developing a bug free - concurrent game (e.g. making sure all users are connected before starting a game, handling disconnect errors and so on).

6.10 Evaluation of WEBGL build

Compiling for WebGL was overall succesful in that it is cross-platform playable on mobile, and desktop with good performance and touch screen responsiveness. However, there was a significant improvement in gameplay using browsers such as Microsoft Edge, Firefox, Safari where Google Chrome performed the worst (with a few bugs).

WebGL allows utilisaing the devices GPU to render 2D and 3D graphics in the web browsewr.

One downside is WebGL 2.0 is deprecated

6.11 Evaluation of Overall Goal and Final Implementation

With reference to the project brief (see Appendix ...), all learning objectives were met, and the end result still held true to the initial goal and scope which was to 'create a multiple accessible game', researched from gamification techniques and mechanics in order for people to use them for the purpose of training.

7 Project Management

The following section describes the tools and techniques used during this project. Table 6 refers to the project management tools, software, languages, APIs and services required to complete this project.

Tools	Description
GitLab	Project management (Boards, Issues and Milestones) & version control
GitHub Desktop	Local software for handling version control
Trello	Lightweight tool for managing daily and weekly tasks
Workona	Chrome extension for streamlining online research into a succinct workplace
Menderley	Reference management for literature research
Microsoft Teams	Communication software for weekly remote meetings
Google Drive	Cloud storage for research, recording minutes and sharing documents
Lucid Chart	Online software for UML diagrams
Unity3D	Game engine for developing for PC and the Web (WebGL)
C#	Primary programming language for Unity
JetBrains Rider	Integrated development environment for C# & Unity
Photon Cloud	Cloud service for hosting multiplayer servers online
PUN2	Multiplayer networking API
ParralelSync	Open source software to clone the Unity development environment
Itch.IO	Secure domain to host WebGL games online

Table 6: Project Management Tools, Software & Services

7.1 Gantt Chart - Phase One

Table 7 illustrates the project planning and research completed during phase one.

Week Beginning	1	2	3	4	5	6	7	8	9	10
Date beginning	Oct 5	Oct 12	Oct 19	Oct 26	Nov 2	Nov 9	Nov 16	Nov 23	Nov 30	Dec 7
Planning										
Project Brief										
Literature Review										
Submit Project Brief		x								
Project Planning										
Identify Hypothesis & Research Question										
Research										
Review Gamification Tool kits										
Research Pre-Existing Cyber Security Games										
Review Game Mechanics										
Brain Storm Game										
Research Tools/Languages/APIs										
Write Progress Report										
Submit Report										x

Table 7: Gantt Chart for Phase One

7.2 Gantt Chart - Phase Two

Tables 8 and 9 illustrate the planned schedule, and the actual outcome during phase 2. The 'implementation' tasks were adapted from the user stories and scheduled in accordance to their MoSCoW priority, as well as their implementation priority. For example, movable units are required before unit abilities can be implemented.

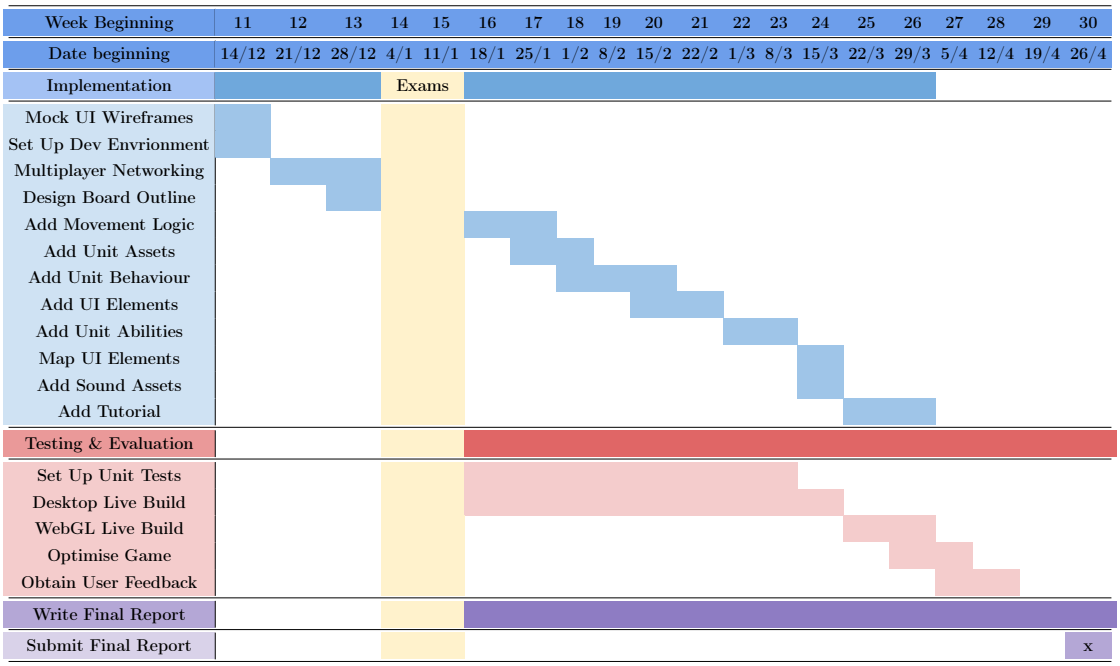


Table 8: Planned Gantt Chart for Phase Two

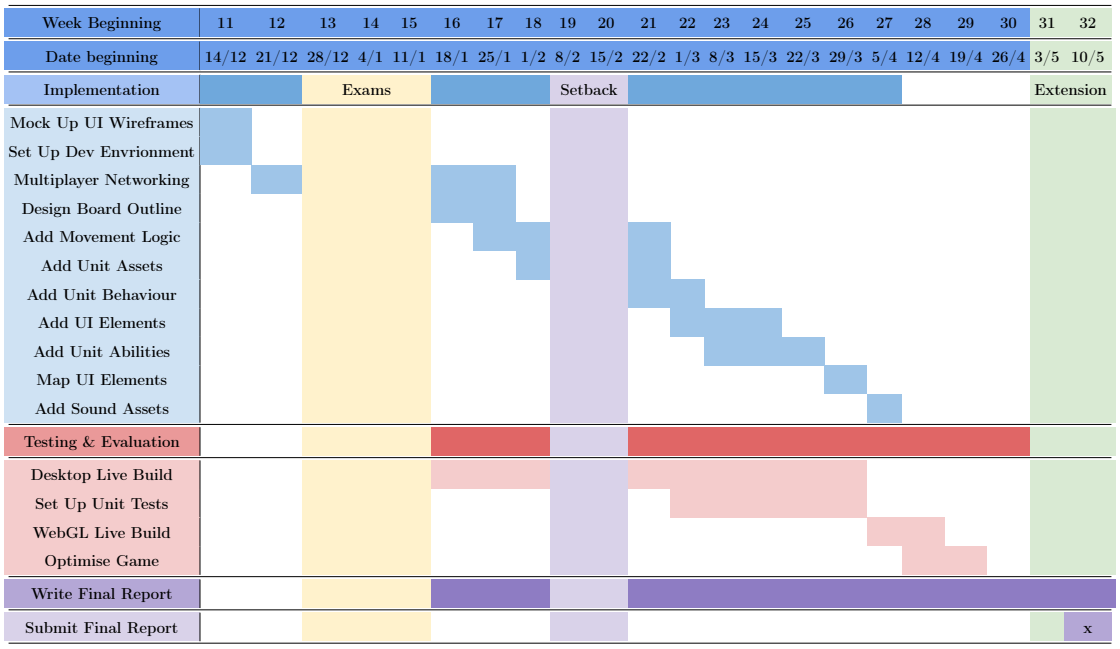


Table 9: Retrospective Gantt Chart for Phase Two

Reflection of Project Management

The project was completed with the Agile methodology in mind. As such, GitLab's Issues and Milestones were used to deliver weekly updates. In turn, this provided invaluable project feedback and assistance which ensured the development process did not deviate.

Despite some personal difficulties in February - March, as all project requirements were completed in accordance to their MoSCoW priority, a minimum viable product was still achieved.

7.3 Risk Assessment

The following risk assessment in table 10 helped to ensure appropriate actions were taken to ensure the successful completion of this project. The risk exposure (R.E) is given by the product of the probability and severity of each risk.

Risk	Prob (1-5)	Sev (1-10)	R.E	Mitigation
Project deadlines not met	3	10	30	Establish soft deadlines and meet with supervisor weekly for continuous feedback & self-evaluation
Online multiplayer failure (limitations/pricing of cloud server hosting)	3	9	27	Develop LAN multiplayer functionality from Unity Mirror API, or local multiplayer co-op
Not obtaining Ethics Approval in time	3	8	24	Refer to generic ERGO extension
Application not effective at teaching	3	7	21	By identifying an appropriate target demographic and the most appropriate cyber security content to teach
Over-estimating scope of implementation	3	7	21	Plan project into smaller iterations and prioritise according to MoSCoW
Application does not relate to original problem statement & hypothesis	2	9	18	Continuously referring back to the initial problem statement and hypothesis
Application does not meet user requirements	2	8	16	Prioritise musts, shoulds then coulds, and functional requirements before non-functional
Sickness, flu or mental health difficulties	2	8	16	Exercising daily, eating healthiy, breaking down tasks into small achievable chunks, and reach out for support if needed
Impact of Covid19	5	3	15	Application will utilise online multiplayer (for remote gameplay), alongside Microsoft Teams, Discord & GitLab for communication and development
Gamified mechanics not implemented appropriately	2	7	14	By identifying key mechanics through literature and game review, and use the MDA framework to decide upon the decided mechanics for the game
Stolen work/data from: cloud storage being compromised & ransomware	1	10	10	Using a secure password manager, 2FA and manually backing up important files on multiple locations
Loss of work/data	1	10	10	Git version control and OneDrive/GoogleDrive for cloud storage
Under-estimating scope of implementation	1	5	5	Further refine/iterate the game after evaluation, and implement all Coulds if time permits

Table 10: Risk Assessment

8 Conclusions and Future Work

8.1 My Contribution

(Began writing up in word, but still needs work)

8.2 Future Work

Additional Scenarios

Evaluation from Feedback

8.3 Final Conclusion

9 Bibliography

[1] M. Bada and J. R. Nurse, "Developing cybersecurity education and awareness programmes for small-and medium-sized enterprises (smes)," *Information & Computer Security*, 2019.

[2] D. Buil-Gil, F. Miró-Llinares, A. Moneva, S. Kemp, and N. Díaz-Castaño, "Cybercrime and shifts in opportunities during covid-19: a preliminary analysis in the uk," *European Societies*, pp. 1–13, 2020.

[3] H. S. Lallie, L. A. Shepherd, J. R. Nurse, A. Erola, G. Epiphaniou, C. Maple, and X. Bellekens, "Cyber security in the age of covid-19: a timeline and analysis of cyber-crime and cyber-attacks during the pandemic," *arXiv preprint arXiv:2006.11929*, 2020.

[4] J.-N. Tioh, M. Mina, and D. W. Jacobson, "Cyber security training a survey of serious games in cyber security," in *2017 IEEE Frontiers in Education Conference (FIE)*. IEEE, 2017, pp. 1–5.

[5] A. T. Forde, "A gamification toolkit for improving cyber security standards adoption," Master's thesis, University of Southampton, September 2020.

[6] R. Hunicke, M. LeBlanc, and R. Zubek, "Mda: A formal approach to game design and game research," in *Proceedings of the AAAI Workshop on Challenges in Game AI*, vol. 4, no. 1. San Jose, CA, 2004, p. 1722.

[7] H. Aldawood and G. Skinner, "Reviewing cyber security social engineering training and awareness programs—pitfalls and ongoing issues," *Future Internet*, vol. 11, no. 3, p. 73, 2019.

[8] J. Abawajy, "User preference of cyber security awareness delivery methods," *Behaviour & Information Technology*, vol. 33, no. 3, pp. 237–248, 2014.

[9] S. Hart, A. Margheri, F. Paci, and V. Sassone, "Riskio: A serious game for cyber security awareness and education," *Computers & Security*, p. 101827, 2020.

[10] A. Jøsang, V. Stray, and H. Rygge, "Threat poker: Gamification of secure agile," in *IFIP World Conference on Information Security Education*. Springer, 2020, pp. 142–155.

[11] J. Anvik, V. Cote, and J. Riehl, "Program wars: a card game for learning programming and cybersecurity concepts," in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 2019, pp. 393–399.

[12] T. Denning, A. Lerner, A. Shostack, and T. Kohno, "Control-alt-hack: the design and evaluation of a card game for computer security awareness and education," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, 2013, pp. 915–928.

[13] C. Koepke, *Persona Image One.* Surface, 2021, Accessed: 19-04-2021. [Online]. Available: https://unsplash.com/photos/E9NcsvbRVqo

[14] Chirstina, *Persona Image Two.* Wocintechat.com, 2021, Accessed: 19-04-2021. [Online]. Available: https://unsplash.com/photos/S3GrMiUhpNU

[15] S. Northcutt, *Persona Image Three.* Unsplash, 2021, Accessed: 19-04-2021. [Online]. Available: https://unsplash.com/photos/sgZX15Da8YE

[16] "Owasp top 10 web application security risks," 2017, Accessed: 20-04-2021. [Online]. Available: https://owasp.org/www-project-top-ten/

[17] G. Miranda, "Unity tutorial - tactics movement," 2017, Accessed: 20-04-2021. [Online]. Available: http://www.gameprogrammingacademy.com/unity-tutorial-tactics-movement/

[18] "Colour Wheel Calculator," 2021, Accessed: 19-04-2021. [Online]. Available: https://www.sessions.edu/color-calculator/

[19] "Web Content Accessibility Guidelines," 2021, Accessed: 19-04-2021. [Online]. Available: https://www.w3.org/TR/2018/REC-WCAG21-20180605/

[20] "Font Contrast Checker," 2021, Accessed: 19-04-2021. [Online]. Available: https://webaim.org/resources/contrastchecker/?fcolor=000000&bcolor=D3CCB3

[21] "Font Contrast Checker," 2021, Accessed: 19-04-2021. [Online]. Available: https://webaim.org/resources/contrastchecker/?fcolor=000000&bcolor=F5F2E9

[22] "Colour blindness awareness," 2021, Accessed: 19-04-2021. [Online]. Available: https://www.colourblindawareness.org/

[23] D. Nichols, "Coloring for colorblindness," 2021, Accessed: 19-04-2021. [Online]. Available: https://davidmathlogic.com/colorblind/#%23E1BE6A-%2340B0A6

[24] "Doodad Pattern Generator," 2021, Accessed: 19-04-2021. [Online]. Available: https://doodad.dev/pattern-generator/

[25] D. Baron, *Hands-On Game Development Patterns with Unity 2019: Create engaging games by using industry-standard design patterns with C.* Packt Publishing Ltd, 2019.

[26] "Unity 3d," 2021, Accessed: 19-04-2021. [Online]. Available: https://unity.com/

[27] "Unity - manual: Getting started with webgl development," 2021, Accessed: 19-04-2021. [Online]. Available: https://docs.unity3d.com/Manual/webgl-gettingstarted.html

[28] "Phaser3," 2021, Accessed: 19-04-2021. [Online]. Available: https://phaser.io/phaser3

[29] "Photon cloud vs photon server," 2021, Accessed: 19-04-2021. [Online]. Available: https://doc.photonengine.com/en-us/realtime/current/getting-started/onpremises-or-saas

[30] "Pun2 documentation," 2021, Accessed: 19-04-2021. [Online]. Available: https://doc-api.photonengine.com/en/PUN/v2/index.html

[31] "Mirror networking," 2021, Accessed: 19-04-2021. [Online]. Available: https://mirror-networking.gitbook.io/docs/

[32] "Rpcs and raiseevents," 2021, Accessed: 19-04-2021. [Online]. Available: https://doc.photonengine.com/en-us/pun/v2/gameplay/rpcsandraiseevent

[33] Foodylag, "Spider Robot," OpenGameArt.org, 2017, Accessed: 28-04-2021. [Online]. Available: https://opengameart.org/content/spider-robot

[34] Lacomap, "Droid," Blendswap.com, 2017, Accessed: 28-04-2021. [Online]. Available: https://www.blendswap.com/blend/19351

[35] Z. Shaheen, "Portal 2 Weatley," Blendswap.com, 2015, Accessed: 28-04-2021. [Online]. Available: https://www.blendswap.com/blend/15053

[36] Galbatorix123, "Cam Bot," Blendswap.com, 2014, Accessed: 28-04-2021. [Online]. Available: https://www.blendswap.com/blend/13529

[37] Rollerrabit2, "Servomotor Simple," Blendswap.com, 2017, Accessed: 28-04-2021. [Online]. Available: https://www.blendswap.com/blend/19199

[38] Reaper2k, "Sci-Fi Thing," Blendswap.com, 2015, Accessed: 28-04-2021. [Online]. Available: https://blendswap.com/blend/14463

[39] "Parralelsync - github," 2021, Accessed: 19-04-2021. [Online]. Available: https://github.com/VeriorPies/ParrelSync

[40] "Unity - manual: The profiler window," 2021, Accessed: 19-04-2021. [Online]. Available: https://docs.unity3d.com/Manual/ProfilerWindow.html

[41] C. Continisio, "Introduction to profiling in Unity 2020," YouTube, 2020, Accessed: 28-04-2021. [Online]. Available: https://www.youtube.com/watch?vūXRURWwabF4&t=431s&ab_channelŪnity

[42] W. Walk, D. Görlich, and M. Barrett, "Design, dynamics, experience (dde): an advancement of the mda framework for game design," in *Game Dynamics*. Springer, 2017, pp. 27–45.

10 Appendices

10.1 Unit Test Cases

Current Defence	Max Defence	Defence Gain	Expected Defence
0	3	0	0
0	3	1	1
1	3	2	3
2	3	3	3
0	0	1	0
3	3	-1	3

Table 11: Testing 'Restore Shields' Function

Defence	Health	Damage	Expected Health	Expected Defence
0	0	0	0	0
1	2	1	1	1
2	2	2	0	2
3	2	3	0	3
3	2	-1	2	3

Table 12: Testing 'Bypass Shields' Function

Defence	Health	Damage	Expected Health	Expected Defence
0	0	0	0	0
0	1	0	0	0
0	1	1	0	0
0	2	1	1	0
1	0	1	0	0
1	1	1	1	0
1	2	2	1	0
2	0	3	0	0
1	1	-1	1	1

Table 13: Testing 'Take Damage' Function

Action Points	Max AP	AP Increment	Expected AP
3	6	-10	3
3	6	-5	3
3	6	0	3
3	6	2	5
3	6	3	6
3	6	10	6

Table 14: Testing 'Increment AP' Function

Action Points	Max AP	AP Decrement	Expected AP
3	6	-10	3
3	6	-5	3
3	6	0	3
3	6	2	1
3	6	3	0
3	6	10	0

Table 15: Testing 'Decrement AP' Function

Action Points	Max AP	AP Requirement	Expected Return
3	6	-10	FALSE
3	6	-5	FALSE
3	6	0	TRUE
3	6	2	TRUE
3	6	3	TRUE
3	6	10	FALSE

Table 16: Check Ability AP Requirement

10.2 Performance Profiling in Unity3D

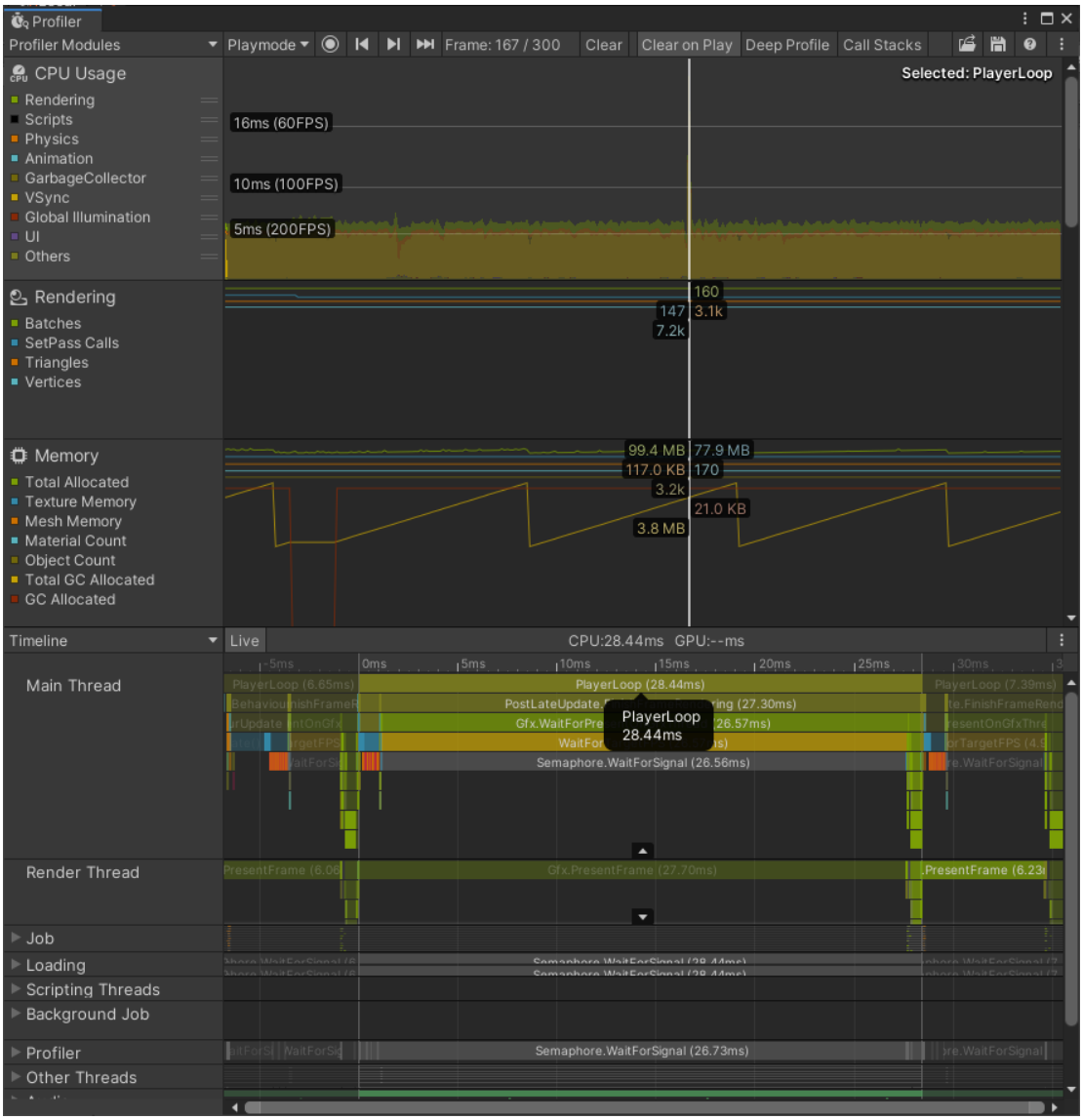


Figure 47: Initial Performance with a Mean Budget (7.09) and Lag Spike of 28.44ms

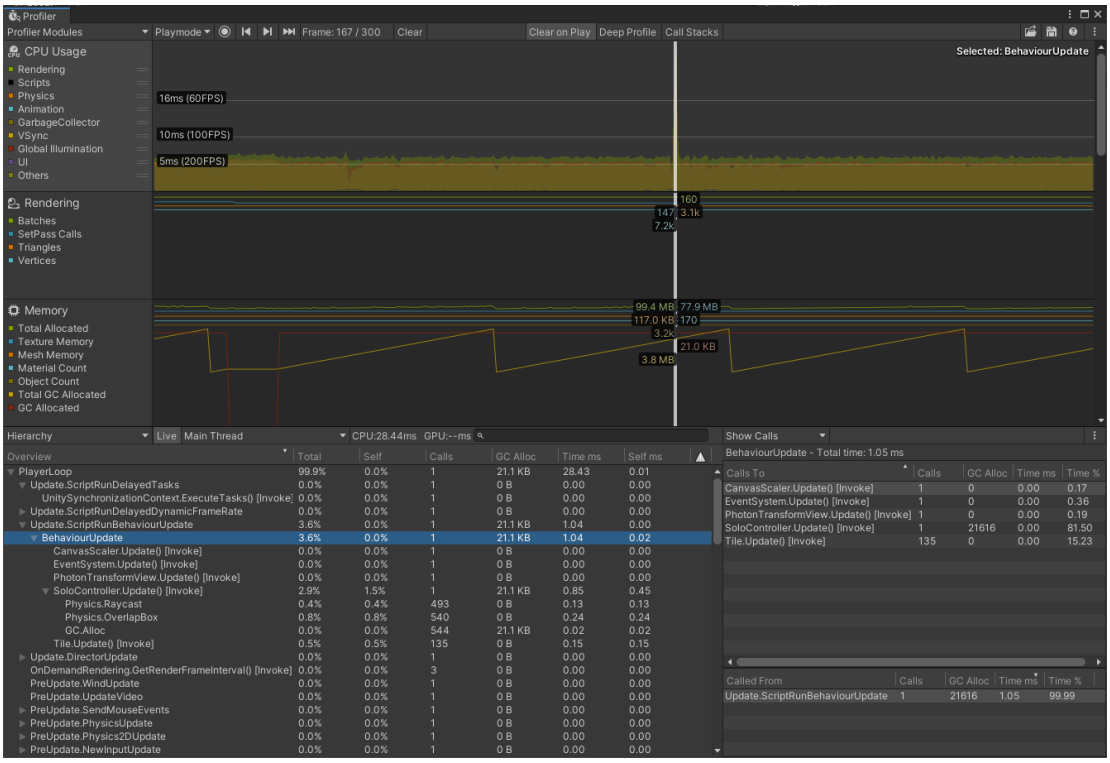


Figure 48: 135 Tile Calls Occupying 15% of the Player's Update Loop

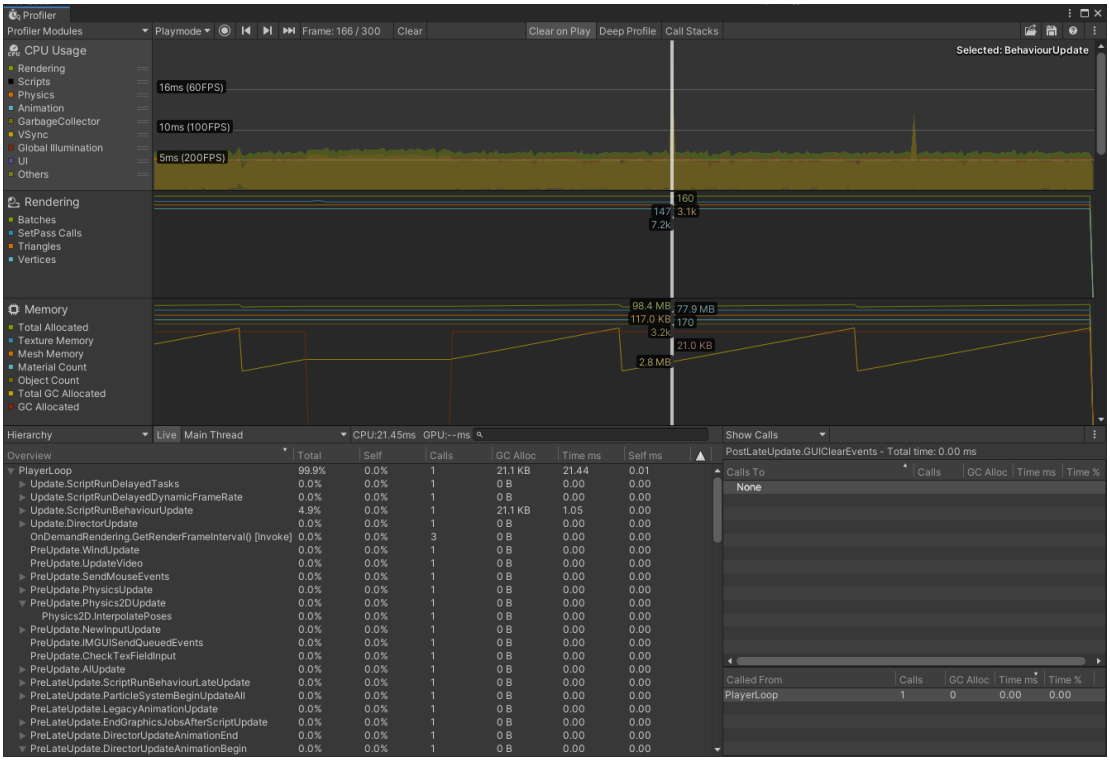


Figure 49: Reduction of lag spike to 21.44ms

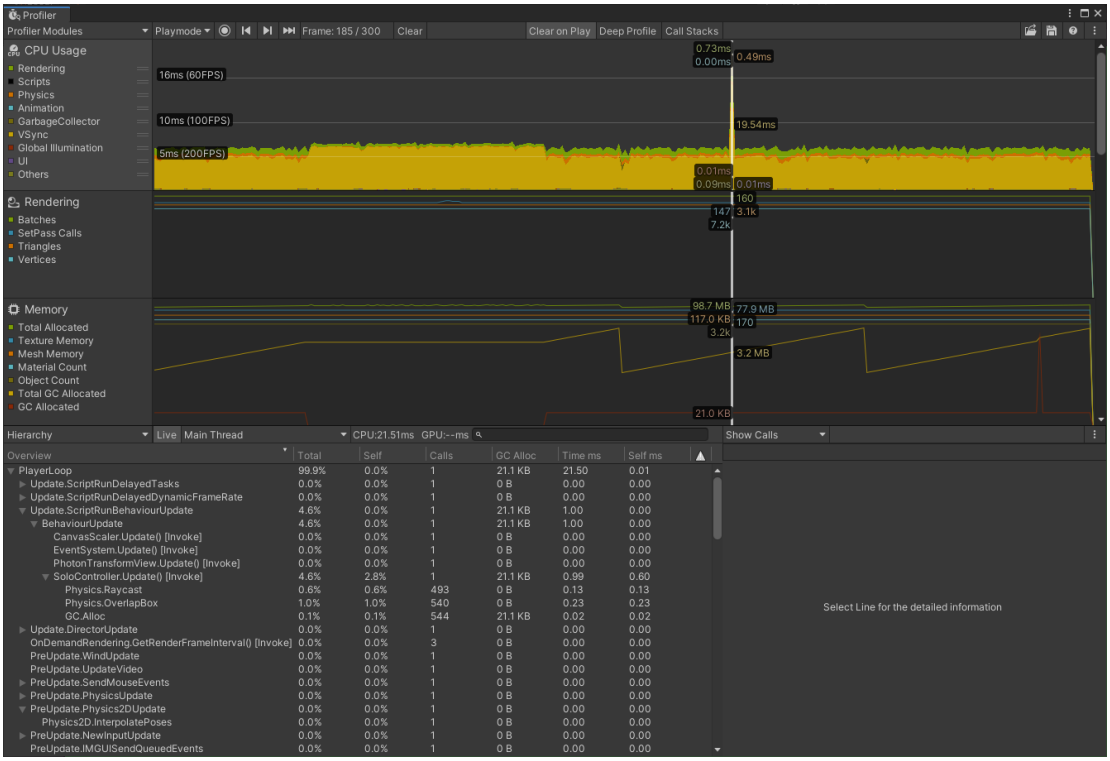


Figure 50: Performance Improvement after Removing Tiles from the Update Loop

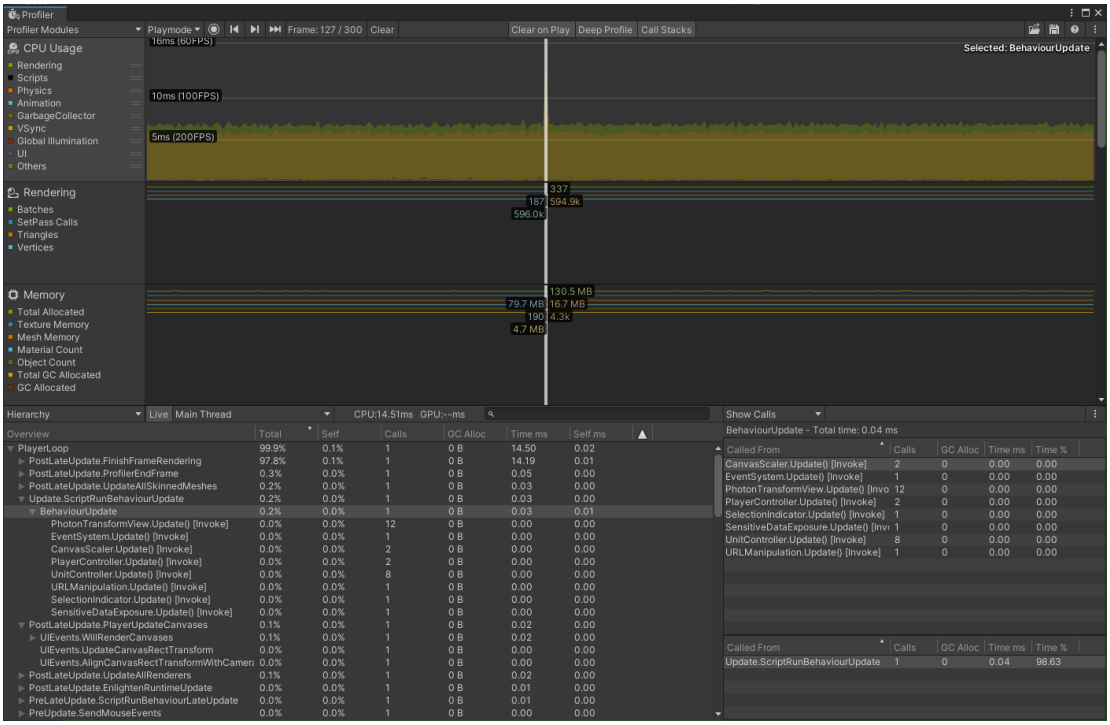


Figure 51: Performance Improvement from Removing Unnecessary Ray-casts